

The copyright of this thesis rests with the University of Cape Town. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

COMPONENT UNIT PRICING THEORY

DAVID WILLIAM CATTELL

Thesis Presented for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Construction Economics and Management

Faculty of Engineering

UNIVERSITY OF CAPE TOWN

June 2009

COMPONENT UNIT PRICING THEORY

DAVID WILLIAM CATTELL

July 2009

-- ABSTRACT --

Building contractors are often commissioned using unit price based contracts. They, nevertheless, often compete on the basis of their overall project bids and yet are paid on the basis of these projects' constituent item prices.

If a contractor decides these prices by way of applying an uneven mark-up to their estimates of their costs, this is known as *unbalanced bidding*.

This research provides proof and explanation that different item pricing scenarios produce different levels of reward for a contractor, whilst exposing them to different degrees of risk. The theory describes the three identified sources of these rewards as well as provides the first explanation of the risks. It has identified the three types of risk involved and provides a model by which both the rewards as well as these risks can now be measured given any item pricing scenario.

The research has included a study of the mainstream microeconomic techniques of Modern Portfolio Theory, Value-at-Risk, as well as Cumulative Prospect Theory that are all suited to making decisions that involve trading-off prospective rewards against risk. These techniques are then incorporated into a model that serves to identify the one item pricing combination that will produce the optimum *value of utility* as will be best suited to a contractor's risk profile.

The research has included the development of software written especially for this purpose in Java so that this theory could be tested on a hypothetical project. A test produced an improvement of more than 150% on the present-value worth of the contractor's profit from this project, if they apply this model compared to if they instead price the project in a balanced manner.

PUBLICATIONS ASSOCIATED WITH THIS THESIS

Parts of the work of this thesis have been published in other media, using the normal student/supervisor protocols. These publications (including two that are in the process of review) include:

Cattell, D.W., Bowen, P.A. and Kaka, A.P. (2004) A model to distribute mark-up amongst quotation component items: an outline, in Proceedings of the 2nd Postgraduate Conference on Construction Industry Development, Cape Town, 10-12th October 2004, Construction Industry Development Board, pp. 154-165. (This paper outlined the overall theory of this thesis.)

Cattell, D.W., Bowen, P.A. and Kaka, A.P. (2007) Review of unbalanced bidding models in construction. Journal of the Construction Division, American Society of Civil Engineers, Vol. 133, No. 8, pp. 562-573. (This paper is based on Chapters 2 and 3 of this thesis.)

Cattell, D.W., Bowen, P.A. and Kaka, A.P. (2008) A simplified unbalanced bidding model. Construction Management and Economics, Vol. 26, No. 12, pp. 1283-1290. (This paper is based on Chapter 4 of this thesis.)

Cattell, D.W., Bowen, P.A. and Kaka, A.P. The risks of unbalanced bidding. Construction Management and Economics. Under review. (This paper is based on Chapter 5 of this thesis.)

Cattell, D.W., Bowen, P.A. and Kaka, A.P. A proposed framework for applying cumulative prospect theory to an unbalanced bidding model. Construction Management and Economics. Under review. (This paper is based on Chapters 6 and 7 of this thesis.)

“I believe in intuition and inspiration. Imagination is more important than knowledge. For knowledge is limited, whereas imagination embraces the entire world, stimulating progress, giving birth to evolution. It is, strictly speaking, a real factor in scientific research.”

Albert Einstein (1931) *Cosmic religion: with other opinions and aphorisms*, New York: Covici and Friede, p. 97.

Acknowledgements

Professor Paul Bowen (Deputy Dean, Faculty of Engineering and the Built Environment) of the Department of Construction Economics and Management, University of Cape Town, for so much, starting from his belief that I was able to return to this research after a long absence. He is an extraordinary academic and I am so grateful for his considerable, untiring efforts keeping me on track as my supervisor.

Professor Ammar Kaka (Deputy Head of the Dubai Campus and the William Watson Chair of Building Engineering and Professor of Construction Economics and Management, Heriot-Watt University, Edinburgh, UK) for his co-supervision and his valuable input.

I am also greatly indebted to Professor Martin Skitmore from the Department of Quantity Surveying, Queensland University of Technology for his encouragement and expert advice. Our frequent exchange of e-mails helped keep me going.

Much of this thesis has already been published as it progressed and in the process an invaluable amount of input was obtained by way of the peer-review process from numerous unknown referees.

Dr Frederik Pretorius, now an Associate Professor at the University of Hong Kong, who many years ago (at the University of the Witwatersrand) drove me into academic research, and the start of this research project. He was the first to instil in me a respect for the process of scientific research as well as the need for discipline whilst recording one's research thoroughly and articulately. He set a standard to which I aspire.

Mr Josh Heymann, a practising quantity surveyor in Sydney, Australia, who served as my initial supervisor at the University of the Witwatersrand, 25 years ago. Josh was Head of Department and backed me when I most needed it, to give me entry into academia at a very young age – sowing the seed of this research.

Finally, and most importantly, I am hugely indebted to my wife, Dr Michelle Lai, who has put up with my many years of starting work at 4 AM and many lost hours and weekends spent on this research.

Thank you.

To my wife, Michelle, and daughter, Megan.

University of Cape Town

I, David William Cattell, hereby...

- (a) grant the University of Cape Town free licence to reproduce the above thesis in whole or in part, for the purpose of research;
- (b) declare that:
 - (i) the above thesis is my own unaided work, both in concept and execution, and that apart from the normal guidance from my supervisors, I have received no assistance except as stated below:
 - Prof Skitmore suggested I consider using Utility Theory
 - Feedback from anonymous reviewers of journal papers
 - (ii) neither the substance nor any part of the above thesis has been submitted in the past, or is being, or is to be submitted for a degree at this University or at any other university.

I am now presenting the thesis for examination for the degree of PhD.

SIGNED: _____

DATE: _____

TABLE OF CONTENTS

	Page
ABSTRACT	i
PUBLICATIONS ASSOCIATED WITH THIS THESIS	ii
QUOTATION	iii
ACKNOWLEDGEMENTS	iv
DEDICATION	v
DECLARATION	vi
CHAPTERS	
1. INTRODUCTION	1
The research problem	3
The research questions	3
The research hypotheses	4
Research aims	5
Research objectives	5
Research methodology	5
Research limitations	6
Thesis structure	6
2. BACKGROUND	9
2.1 Introduction	9
2.2 Ethics	12
2.3 Conclusion	18
3. OTHER UNBALANCED BIDDING MODELS	22
3.1 Introduction	22
3.2 Gates' strategy	24
3.3 Stark's model	27
3.4 Ashley and Teicholz's models	32
3.5 Diekmann, Mayer and Stark's models	41
3.6 Tong and Lu's model	46
3.7 Afshar and Amiri's model	47
3.8 Christodoulou's model	48
3.9 Conclusion	50

4.	MODELLING A PROJECT'S REVENUE	56
	4.1 Introduction	56
	4.2 The ranking of items	57
	4.3 Front-end loading	58
	4.4 Individual rate loading	60
	4.5 Back-end loading	62
	4.6 Complex composite loading	64
	4.7 Tender price constraint	68
	4.8 Price rate constraints	68
	4.9 Summary	70
5.	MODELLING A PROJECT'S RISK	73
	5.1 Introduction	73
	5.2 The risks of item pricing	74
	5.3 Two forms of risk	81
	5.4 Value-at-risk	82
	5.5 Valuing these risks	84
	5.6 Discussion	93
	5.7 Summary	93
6.	MODERN PORTFOLIO THEORY	98
	6.1 Introduction	98
	6.2 Markowitz	100
	6.3 Applied to unit pricing	103
7.	CUMULATIVE PROSPECT THEORY	106
	7.1 Introduction	106
	7.2 An efficient frontier	106
	7.3 Subjective perspective on risk	107
	7.4 Utility theory	108
	7.5 Expected utility theory	109
	7.6 Prospect theory	113
	7.7 Cumulative prospect theory	115
	7.8 Practical application of the theory	117
	7.9 Summary	118
8.	SOFTWARE FOR TESTING THE MODEL	123
	8.1 Introduction	123
	8.2 The Java technology platform	124
	8.3 Overview	126
	8.4 The program	134
9.	TESTING THE MODEL	157

10.	CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER WORK	167
	10.1 The findings	169
	10.2 The research hypotheses	172
	10.3 Conclusions	174
	10.4 Recommendations for further research	174
	10.5 Research aims and objectives	177
	APPENDICES	178
A.	SOFTWARE	178
B.	HYPOTHETICAL PROJECT	239
C.	TEST RESULTS	260
D.	SYSTEM FEEDBACK	262
E.	ITEM PRICES	265

University of Cape Town

1. INTRODUCTION

In the construction industry, with unit price contracting, contractors compete on the basis of their bids for whole, composite projects. These projects comprise many hundreds or thousands of component items, often described in detail in bills of quantities. In the UK method of contracting (popular across the Commonwealth countries), it is typical when contractors compete by way of tendering, that clients will provisionally select one contractor from all of the bids submitted to them. This is often the lowest bidder, although not necessarily so. The client will then request that this contractor submit their priced bills of quantities, as preparation for potentially awarding them the contract. At this stage, the contractor has to decide their prices for each unit of each of the project's constituent items. This process is known as "item pricing" or "unit pricing".

Significantly, the item prices are not subject to the same competitive restraint as the pricing for the overall project. Nevertheless, it is these item prices, rather than the tender price, that will govern the contract. These item prices will be used to determine the monthly interim payments to the contractor as they progress; they will also be used to determine the quantum of escalation compensation (for inflationary increases in costs); and they will be used to value the variations to the design of the project (as is very common).

Ordinarily, one might expect a project tender price to be generated as a summation or aggregation of its constituent component item prices, but the reverse is also possible. It is, of course, also possible that the contractor can choose from many millions of different item price combinations which all summate to the same combined tender price. These different pricing scenarios will deliver different benefits to the contractor.

Unbalanced bidding models are mathematical tools for use to determine the optimum unit prices. In the past, these models have been largely focused on the optimization of the expected profits for a contractor rather than give much consideration to the risks

involved. Researchers (see Gates, 1967; Stark, 1968, 1972, 1974) have, however, often acknowledged that there are risks but, despite this, they have made little effort to properly incorporate these risks into their models.

Moreover, little has been done to structure these models so that they recognize the inherent nature of the trade-off that exists between unbalanced bidding's contributions to these risks and that of the prospective gains.

All models (for examples, see Stark, 1968, 1972, 1974; Teicholz and Ashley, 1978; Diekmann *et al.*, 1982; Tong and Lu, 1992; and Christodoulou, 2008) have otherwise given recognition to risk by constraining prices using an imposition of lower and upper bounds to each and every item price. Although this does not seem to have been intended, these bounds end up becoming the single most important aspect of these models. The effect is that all items are priced at either their upper price limit or else their lower price limit, with the exception of only one item. This one remaining item price then serves to ensure that the summation of all the priced items equals the tender price. The effect of these models is therefore reduced to only serving to split all the items into these two subsets: those priced high (and assigned the high price that is to be chosen arbitrarily by the contractor) and those priced low (and similarly, assigned the contractor's chosen low price). These models are, therefore, heavily dependent on the contractor's choice of high and low prices and yet this does not appear to have been given much consideration when these models were formulated.

This thesis will show that whilst it is heuristically appropriate to constrain the prices for all items, it is not appropriate that these constraints be imposed as fixed, non-negotiable limits.

This thesis describes a theory as regards item pricing in which the pursuit of profits and the avoidance of risk are treated as being of equal significance. Furthermore, it will be shown that contractors can manage these two objectives collectively, such that they are able to pursue the optimization of a personalised *value of utility* (representing a trade-off, best suited to them, between striving for high profits whilst avoiding excessive risk).

THE RESEARCH PROBLEM

Despite the passage of 50 years since research on unbalanced bidding was first recorded, unbalanced bidding models have not yet been adopted for practical use. These models have focussed on the maximisation of profit and have given little consideration to the risks involved. The past research has, however, acknowledged that there are considerable risks, but these risks have largely not been identified, let alone modelled. These models have also typically focussed on maximising only one or other of the benefits from item price loading, and have not provided a comprehensive approach to address all of the potential opportunities for improving profits.

The problem to be researched is therefore stated to be as follows:

Unbalanced bidding models have not addressed the risks of an uneven mark-up despite this having been identified as an important consideration at the outset of research in this field, 50 years ago. These models have also not quantified all of the various sources of improved profits. Overall, unbalanced bidding models have so-far failed to provide contractors with a meaningful or useful technique for optimising their item pricing with respect to profit and risk.

THE RESEARCH QUESTIONS

The research questions are formulated as follows:

- (1) *What are the benefits that can be derived from an uneven distribution of mark-up amongst a project's constituent items?*
- (2) *Is it possible that one model could comprehensively and collectively quantify all of these benefits?*
- (3) *What are the risks that contractor exposes themselves to, in the event that they price items without using a consistent mark-up?*
- (4) *Is a contractor able to quantify these risks, for any specific item price combination?*

- (5) *If it is possible for a contractor to quantify both the prospective profits as well as the risks for a comprehensive range of different item price combinations, how can a contractor choose from these to find the one best suited to them?*

Subsidiary questions to be addressed include:

- (1) *What other unbalanced bidding models have been proposed and how have these succeeded / failed to provide an effective solution?*
- (2) *What has research revealed that has been conducted in other areas of microeconomics as regards decisions that entail trading-off returns against risk?*
- (3) *What are the ethical considerations as regards whether it is acceptable for contractors to price items using different mark-ups and are they ethically obliged to price all items by way of using the same mark-up?*

THE RESEARCH HYPOTHESES

This thesis sets out to provide proof of the following three hypotheses:

- (1) *The uneven allocation of mark-up between component items can have a significant effect on both the profitability and the risk of a project.*
- (2) *Some item price combinations may be considered more **efficient** than others: some such combinations will contribute the most expected profit for the same or lesser degree of risk than other combinations. On this basis, it is not rational for a contractor to choose to use any prices that are not **efficient**.*
- (3) *A contractor is able to identify the item prices that will deliver the best compromise for them between risk and reward, as judged in accordance with their personal attitude to risk.*

RESEARCH AIMS

The intended aim of this research is as follows:

To establish a scientific basis for more-effective, better-informed item pricing by contractors.

RESEARCH OBJECTIVES

The objectives of this research are as follows:

- (1) to extend the present theoretical foundation for component item pricing,*
- (2) in particular, to gain better insight into both the risks as well as the rewards generated from item pricing, and*
- (3) to establish a new mathematical model that quantifies both the risks and rewards of item pricing - to facilitate the identification of item prices that will give effect to a compromise between the pursuit of rewards, together with the restraint required to avoid excessive risk, as is suited to the circumstance and psychology of a particular contractor.*

RESEARCH METHODOLOGY

The methodology for the research includes the following:

- (1) A literature review of other unbalanced bidding models since their conception 50 years ago.*
- (2) The identification of the rewards that item pricing is able to provide.*
- (3) The mathematical modelling of these rewards.*
- (4) The identification of the risks that item pricing gives exposure to.*
- (5) The mathematical modelling of these risks.*

- (6) *A literature review of relevant mainstream microeconomic theory as regards decision-making entailing trading off returns and risks.*
- (7) *An application of a choice of technique (identified by way of this review), for combining consideration of the rewards and risks generated by different item pricing regimens.*
- (8) *The development of a software system by which to implement these models.*
- (9) *Testing this system using the data from a small hypothetical project and assessing the results, in particular as to whether they provide proof of the research hypotheses.*

SCOPE OF THE WORK

This research is limited to consideration of the pricing executed by building contractors, engaged in the construction industry in South Africa, who procure work by way of tendering, and in particular, where clients have chosen to make use of bills of quantities as part of the contract documentation. South Africa's construction industry has inherited its practices from the United Kingdom and these are largely shared by many other countries that are members of the Commonwealth of Nations. The applicability of this research though is also believed to apply to civil engineering work as well as other industries such as the oil and forestry industries, which are known to use unit price based contracts and where unbalanced bidding is also prevalent.

THESIS STRUCTURE

Following this introduction, **Chapter 2** will provide background information to the practice of item pricing, in particular as regards different techniques of "item price loading" and also the ethical concerns with respect to unbalanced bidding.

Chapter 3 will provide a critical review of all known prior research on unbalanced bidding, showing that little has been accomplished since the 1950s (when it was first

identified as having significant potential) as regards equipping contractors with practical techniques for item pricing.

Chapter 4 will provide an analysis of the benefits that can be generated from unbalanced bidding and it will provide a model by which these can be assessed.

Chapter 5 will propose another model (in addition to the one proposed in Chapter 4) in which the “risks” of item pricing are identified and assessed.

Chapters 6 and 7 describe relevant mainstream microeconomic theories as regards methods of managing situations that entail trading-off profits against risk. In particular, these chapters provide an introductory description of Modern Portfolio Theory and Cumulative Prospect Theory, respectively. These techniques provide a basis by which the contractor can identify the pricing combination that will deliver their optimum *value of utility*, representing their best, personal compromise between profit and risk.

The composite new theory as regards Component Unit Pricing is then put to the test.

Chapter 8 describes software that has been written in Java especially for the purposes of applying this theory using the data of a small hypothetical project. This Java software incorporates Monte Carlo simulation that is aided by a hybrid combination of techniques, incorporating aspects of artificial intelligence, genetic computing, and fuzzy logic.

Chapter 9 describes the results of these tests of the composite model. It provides proof of the research hypotheses and indicates that this line of research has the potential to be significant to contractors and that further research is justified.

The concluding discussion is provided in **Chapter 10**, together with suggestions for further research in this field.

Appendix A provides a listing of the source code of the software written to implement the model and **Appendix B** shows the data that describes the small hypothetical project that is used as the basis of the test. **Appendices C – E** show various output from this test (which is explained in Chapter 9).

REFERENCES

- Christodoulou, S.E.** (2008) A bid-unbalancing method for lowering a contractor's financial risk. *Construction Management and Economics*, no. 26, pp. 1291-1302.
- Diekmann, J.E., Mayer, R.H. Jr. and Stark, R.M.** (1982) Coping with uncertainty in unit price contracting. *Journal of the Construction Division, American Society of Civil Engineers*, vol. 108, no. C03, Sep, pp. 379-389.
- Stark, R.M.** (1968) Unbalanced bidding models – theory. *Journal of the Construction Division, American Society of Civil Engineers*, vol. 94, no. C02, pp. 197-209.
- Stark, R.M.** (1972) Unbalancing of tenders. *Proceedings of the Institute of Civil Engineers*, Technical Note 59, vol. 51, pp. 391-392.
- Stark, R.M.** (1974) Unbalanced highway contract tendering. *Operational Research Quarterly*, vol. 25, no. 3, pp. 373-388.
- Teicholz, P.M. and Ashley, D.B.** (1978) Optimal bid prices for unit price contract. *Journal of the Construction Division, American Society of Civil Engineers*, vol. 104, no. 1, March, pp. 57-67.
- Tong, Y. and Lu, Y.** (1992) Unbalanced bidding on contracts with variation trends in client-provided quantities. *Construction Management and Economics*, no. 10, pp. 69-80.

2. BACKGROUND

2.1 INTRODUCTION

This chapter will provide some background to this topic before the next chapter details the history of research in this field. This discussion includes a consideration of the client's perspective on this practice, including an analysis of the view that unbalanced bidding is unethical.

Bidding models are mathematical techniques designed for use by building contractors, amongst others, to assist them with optimising their bid prices in competitive tenders. This area of research has been led by that of Friedman (1956) and Gates (1959) and more than 1000 papers have been published since then with much of the debate focussed on the underlying mathematics. The debate has gone on for 40 years and it is only recently that Skitmore *et al.* (2007) has provided a proof that the mathematics advocated by Gates is to be preferred over that which was instead proposed by Friedman. However, Skitmore (2004) warns that both methods are problematic.

This style of 'bidding model' is focussed on the determination of a project's overall bid price. Contractors are also, however, required to submit prices for each constituent component item of projects in such a manner that the summation of these unit prices equates to the overall project price. These item prices are then used as the basis by which these contracts are then administered.

Gates (1959) was the first to identify the role of item price loading as a tendering strategy. This approach entails allocating different mark-ups to individual items within a project so as to realise advantages that are not likely to be accomplished by way of allocating a universally constant mark-up to all of a project's items. Further research (see, for example, Stark, 1968; Diekmann *et al.*, 1982; Ashley and Teicholz, 1977; Tong and Lu, 1992), reviewed in Chapter 3, led to the development of a variety of

mathematical techniques by which to optimise this and these have become known as *unbalanced bidding models*. They are very different from what is popular known simply as ‘bidding models’ (as described above) and they are not to be confused.

Unbalanced bidding models are typically designed for use by building or engineering contractors and are often utilized in the oil and forestry industries (see Athey and Levin, 2001). This study is, however, concerned primarily with the use of these models by contractors in the construction industry although much of this research is relevant to the full spectrum of the potential use of these models.

Common to these models (such as those advocated by Gates, 1959, 1967; Stark, 1968, 1972, 1974; Ashley and Teicholz, 1977; Teicholz and Ashley, 1978; Diekmann *et al.*, 1982; Cattell, 1987; and, Tong and Lu, 1992) is that their objective is to maximize the present-day value of a project’s profit. They all entail the “loading” of the prices of some items, and the “unloading” of the prices of other items, in this endeavour.

Many contractors are said to avail themselves of this opportunity, according to McCaffer, (1979), Green (1986), Kaka and Price (1991) and Kenley (2003).

The effects of item price loading

For the most part, each item of work within a project has largely different characteristics to other such items. Some relate to work that has to be done early in the construction schedule; others to activities scheduled later. Many fall within different escalation workgroups in terms of contract price adjustment provisions. Some items have an initial quantity attached to them in the bills of quantities that the contractor can be fully confident will not differ from the final quantity. Some others describe work that is expected to vary in quantity when it is built (being measured as ‘provisional’): some of these items may be expected to finally be allocated a higher quantity and others a lower quantity. Some items’ final quantities are easier to estimate than others, and thus some enjoy a higher degree of confidence as regards their variability than others. And so on. Thus, if one considers that each item incorporates many different characteristics in

different proportions; most items are different in their overall character from any other item.

Item price loading as a theory relies on this reality that most items are in many ways unique in their character. By allocating some items higher mark-ups than other items, item price loading is seeking to take advantage of each item's unique attributes.

Consider the following examples:

- If high prices are allocated to items scheduled to arise early in the contract's project plan, the contractor will receive larger amounts of money for the first few interim payments which will aid their initial cash flow for the contract – a practice known as 'front-end loading';
- If the contractor was to allocate high prices to items that are scheduled to occur late in the project plan and to those that fall into workgroups that have a high expected escalation, the contractor will receive larger amounts in escalation in compensation for inflation – perhaps more than the cost to them of escalation – but, more importantly, most certainly more than if they were to allocate lower prices to such items – a practice known as 'back-end loading';
- If a contractor was able to predict variations in the contract's design or identify mistakes in the measured quantities, they may take advantage by allocating high prices to items for which they expect the quantity to be adjusted upwards and low prices to items that they expect will be reduced – a practice known as 'quantity error exploitation'.

Pursuing any one of these opportunities (as examples) in isolation is intuitively very simple. However, the reality is a lot more complex when one considers that each single item within a project does not only have one such characteristic, but instead, they all have a largely unique and complex blend of many such characteristics. Each item cannot simply be described, for instance, as being an 'early' or 'late' item without also having to recognize that it will have other characteristics as well. Each of these characteristics calls

for a different treatment as regards item price loading. For instance, a 'late' item might fall into an escalation workgroup that has a high expected rate of escalation. In this instance, it is not obvious whether this item should be allocated a high or low mark-up. The objectives of front-end loading would suggest heuristically that 'late' items should be allocated low prices and yet the objectives of back-end loading suggest that a 'late' item in a 'high-escalation' workgroup should be allocated a high price. Which of these two is, in this instance, of greater significance?

To add further to the complexity, consider that a project typically comprises many hundreds or thousands of items, most of which are unique in character. Thus, when viewed holistically, with all of the complexity that this has inherent in it, it becomes a lot less obvious how best to pursue the advantages of item price loading. Nevertheless, not only does it remain obvious that these advantages remain intact regardless of this complexity, but also, it is hypothesized that the more complex a project is, the more there are opportunities for item price loading, and the more should be the advantages of this practice.

The complexity of this problem becomes more intricate and demanding when one considers that contractors are not only potentially able to enjoy greater profits from this manipulation of their prices but that this practice typically also exposes them to additional risk. It is, however, possible that contractors can use unbalanced bidding as a technique by which to reduce their risk.

2.2 ETHICS

The objective of unbalanced bidding is for a contractor to derive some advantage that they would not enjoy from a balanced bid. (A balanced bid is one where all item prices are derived using the same, average, mark-up.) This will often, but not always, occur at the expense of the client - most probably without the client initially being aware that they are being given this exposure. Contractors have the advantage over clients in this regard by having available to them their own confidential information about their estimates of a project's cost. Clients do not have access to this same information, which makes it far

more difficult for them to determine the existence and / or the extent of any possible loading in any item prices.

Contractually, clients will typically, in terms of traditional building practice, provide (by way of their professional team) a specification and design of the project that they wish to have built. Contractors commit themselves legally to fulfilling these requirements, but they are left entitled to decide their own approach and technique by which they will accomplish this. This power of discretion that is enjoyed by contractors is significant here from the perspective that different contractors are entitled to choose different methods of construction that might lead to different costs of production. Furthermore, different contractors have different resources available to them, again with corresponding differences in cost. For example, one contractor might have available for a project some plant, such as a crane, which may be very well suited to a project and in which their long-standing investment is already 'written-off' in their books giving them a cost advantage over their competitors. Different contractors have different suppliers and sub-contractors or at least might have different relationships with these. Thus, different contractors will incur different costs if they were all to build the same project, even if they were to build it at the exact same time, with the exact same set of imposed conditions (such as the weather). Contractors also differ from one another by way of the techniques that they use to estimate these costs and thus different contractors, whilst competing for the same job, might reasonably be expected to have estimated item costs that are quite different from one another. Research by Beeston (1975) has shown that the variance between contractors in individual, component, estimated item costs far exceeds the overall variance that lies between contractor's overall, composite, estimated project costs.

With these variances being in existence, a client might have difficulty being able to differentiate between a balanced set of item prices and an unbalanced one. Furthermore, if one recognizes from the above that contractors are not all operating off the exact same set of estimated costs, why is it ethically significant that they should be expected to present a balanced bid? Should contractors not be entitled to price different items with different mark-ups if only so as simply to recognize their relative, competitive differences from other contractors, with respect to any such items? For example, if their estimated

cost of their crane were extremely low, perhaps for reason that it was purchased many years ago, could it not be justified that they might apply a high mark-up to their unusually-low cost? Is there, indeed, any standard method by which they should feel obliged to account for their costs, especially as regards more ambiguous issues such as the use of depreciating assets and overheads? If a contractor uses a method of construction that is unique to them, that leads to different costs than other contractors, should they account for this on the basis of these differences or on the basis of the industry norm? How should they recoup their costs of research and development in such efforts, and how should they account for the costs of training their staff, especially if this has given them unique advantage by way of productivity? Clearly, different contractor's situations need to allow them a freedom by which to account for their internal expenditure in such a way that it respects their independence in an economic environment governed by free-enterprise.

A further consideration as regards the ethics of unbalanced bidding is that a client is given a full disclosure of a contractor's item prices. The contractor's pricing is fully transparent, to the degree that this is requested from clients. Clients are hence given this disclosure on their terms, which includes that they then have the opportunity by which they can exercise the choice of outright rejecting these prices, else of insisting upon an acceptance of the contractor's bid on condition that these prices be renegotiated. Clients have these options on their terms and are able to legitimately choose to rather initiate the same negotiations with one of the other bidders. With these arrangements, contractors are not in a position to force clients to have to accept their unit prices, nor are they able to withhold information as regards these, from the client.

When a contractor submits their item prices they do so without any inherent assurance that these prices constitute a balanced bid. They are asked to present their prices and not their calculations by which they have arrived at these prices. When submitting these prices they do so knowing the risk that they may be found to be unacceptable. They could be rejected as much for reason that they were derived from a peculiar cost estimate as for reason that they are derived from an uneven distribution of mark-up.

As much as a client has no right to dictate to a contractor how they should build a project or how they should estimate their costs, it follows that they have no right to dictate to a contractor how they should price the project either. The client does, however, retain the right to reject outright any prices that a contractor submits to them. To this end, contractors need to consider this risk when pricing a project.

It is furthermore noteworthy that unbalancing bidding does not necessarily imply that the resultant item prices will be extreme as to constitute pricing that is out of norm with what is customary in practice. It is instead to be noted that unbalanced bidding may very well already have become the norm (Kenley, 2003), albeit that item pricing is not known to be practiced in industry using any mathematical models that are designed to optimize this process (see Green, 1986). *Optimisation* does not imply any greater degree of loading, especially if the objective of any such model is also to minimize the associated risks.

Kenley (2003) has questioned the morality of item price loading but nevertheless has acknowledged it as having considerable impact as well as having widespread use. He has, however, found that its use is with limited levels of mathematical sophistication. Green (1989) has likewise commented on the significant value of unbalancing, and in particular from the practice of front-end loading. He identified (Green, 1986) extensive use of what he called 'individual rate loading' and 'front-end loading' although he noted that, in practice, it was conducted in an unscientific manner. Kaka and Price (1991) have also commented on the 'significant' effect of front-end loading in their efforts to account for this practice when they forecast developers' cashflows.

Warning

Kenley (2003) has warned of the dangers to contractors of pursuing unbalanced bidding. This is largely founded on his experience in which some contractors' contracts managers have lost track of the extent or nature of their own estimators' initial price loading.

Kenley has found that there is often poor communication internally between a contractors' estimators and their contracts managers. He found that this has at times given rise to these contracts managers enjoying a false sense of optimism early on in projects when they find it surprisingly easy to contain a project's costs far below the

prices in the bills of quantities. This is said to sometimes give rise to the projection of this perception of good fortune through to the end of a project. Kenley has quoted examples of where these initial periods of surplus cashflow have led to the vast expansion of some contractors' overheads to the extent that some have indulged themselves with excessive, expensive luxuries.

He observed that problems obviously then arise later when these managers are seemingly unable to comprehend why items built later in these projects begin to suffer from increasing lesser profit margins to the extent that work done near the end of these projects is typically having to be built at a loss. His research identified numerous examples of contractors having failed altogether and long established businesses having gone bankrupt due to these short-comings in their cashflow management.

Discussions with Kenley have clarified that these problems should not be blamed on the practice of unbalanced bidding (nor on other judicious 'cash farming' techniques that he has identified). Instead, it is more reasonable to blame this on poor communication within these contractor's management teams as well as on their poor and unsophisticated systems which lack the ability to keep contracts management suitably informed. Kenley has argued that the survival of contractors is highly dependent on these sophisticated systems and management techniques with regards to their cashflow management, especially in an environment where this is made more challenging through the widespread use of item price loading.

The role of the professional quantity surveyor

Kenley (2003: 233) has described the practice of item price loading as 'ethically questionable', 'dubious' and 'illegitimate'. Whilst many researchers have largely steered clear of such judgement, the opinion of Stark (1968 and 1974) is that unbalanced bidding provides contractors with an efficiency that is healthy not only for them but for the industry as a whole as well as for clients. This argument appears to be founded on the logic that the process of tendering is there to serve the purpose of ensuring that the most efficient contractor wins, something which, upon a superficial assessment at least, would appear to clearly benefit clients. The further pursuit of this argument would appear to

suggest that contractors who fail to utilise item price loading will be failing to derive a considerable profitable opportunity that is inherent within the prospective benefits of any project. The consequence of this is then that any contractor not practicing item price loading will be suffering an 'opportunity cost' - thereby effectively rendering them uncompetitive within any environment in which item price loading has become the norm.

Another perspective on this would suggest that, although item price loading might provide increased efficiency for contractors, it generates considerable additional 'hidden costs' for clients (Kenley, 2003). Thus whilst competition within an environment in which unbalanced bidding may be commonplace, may result in lower tender prices, these lower bids may not necessarily represent lower effective costs for the clients.

Nevertheless, whilst item price loading has considerable benefits to offer, it appears illogical to expect rational contractors not to pursue this opportunity. Once doing so, it would appear sensible that any rational contractor would furthermore wish to refine this process to the extent that they are able to maximise this opportunity. It is likewise logical that any contractors not doing the same would be rendered uncompetitive. Thus, the optimised practice of item price loading might become the norm. In this scenario it would be naive for clients to expect anything other than unbalanced bids and thus they will have need to factor the 'hidden costs' that are associated with such bids into their budgets. The role of the quantity surveyor in protecting their client will become more challenging as the practice of unbalanced bidding becomes more widely practiced, more sophisticated, and more extreme. Furthermore, sophisticated item pricing loading models are normally designed to optimally exploit any errors in the bills of quantities caused by the quantity surveyor and thus it may be argued that, within this scenario, it will become more important for clients to invest in a good quantity surveyor as well as in a well prepared bills of quantities.

A possible means by which client might protect themselves from the effects of item price loading would be for them to insist that all bidding contractors are to submit, for comparison, a fully priced bills of quantities. This would facilitate that the quantity surveyor could be able to compare bids not solely on the basis of their bottom-line tender price but rather by comparison of the present value of the anticipated cashflows that each

contractor's priced component items is likely to generate. The quantity surveyor would furthermore be able to run sensitivity analyses on these priced component items, testing for variations in item quantities and thereby should be able to highlight cases of any quantity error exploitation.

Another method by which a quantity surveyor might be able to circumvent and / or moderate some of the benefits of unbalanced bidding is for clients to insist that contractors bids also incorporate a commitment as regards their cashflow drawdown (Kenley, 2003). Interim payments would therefore be derived from this agreed schedule rather than from the use of item prices. As Kenley has proposed, this might necessitate that the client would want to pre-select the contractors who will participate in such a tender, so as to be assured of their reputation and sound standing. If quantity surveyors were to advocate this practice, they would have helped remove much of the incentive for unbalanced bidding. By doing so, the quantity surveyor might hope for more balanced bids, and therefore might expect to have alleviated much of the client's risk.

Clearly, the quantity surveyor has a significant role to play to help protect clients from being exposed to both much of the cost that they would incur from unbalanced bidding as well as the risks that are involved.

2.3 SUMMARY

This chapter has shown that unbalanced bidding models are not to be confused with the far more popular area of research that is described simply as 'bidding models'. It has described how the research on unbalanced bidding arose as an additional strategy related to 'bidding models' and how these two fields of research have never subsequently 'overlapped' with each other.

This background has also included a brief assessment of the impact of this practice on the clients of construction projects and debated some of the ethical issues that are involved that are considered very contentious.

Lastly, it has investigated an overview of the role of the quantity surveyor in this domain and suggested some methods by which they might be able to protect their clients from abuse by way of unbalanced bids.

The next chapter provides the history of research in this field and it provides a review of all known significant unbalanced bidding models that have been proposed in the 50 years since this area of research was initiated.

University of Cape Town

REFERENCES

- Athey, S. and Levin, J.** (2001) Information and competition in US Forest Service timber auctions. *Journal of Political Economy*, vol. 109, no. 2, pp. 375-417.
- Ashley, D.B. and Teicholz, P.M.** (1977) Pre-estimate cash flow analysis. *Journal of the Construction Division, American Society of Civil Engineers*, Proc. Paper 13213, vol. 103, no. C03, pp. 369-379.
- Beeston, D.T.** (1975) One statistician's view of estimating. *Chartered Surveyor, Building and Quantity Surveying Quarterly*, vol. 12, no. 4, pp. 49-54.
- Cattell, D.W.** (1987). Item price loading. *PACE'87 Progress in Architecture, Construction and Engineering proceedings*, Johannesburg, July, Vol.II, Session 3, pp. 1-20.
- Diekmann, J.E., Mayer, R.H. Jr. and Stark, R.M.** (1982) Coping with uncertainty in unit price contracting. *American Society of Civil Engineers, Journal of the Construction Division*, vol. 108, no. C03, Sep, pp. 379-389.
- Friedman, L.** (1956) A competitive bidding strategy. *Operations Research*, vol. 4, no. 1, Feb., pp. 104-112.
- Gates, M.** (1959) Aspects of competitive bidding. *Annual Report*, Connecticut Society of Civil Engineers.
- Gates, M.** (1967) Bidding strategies and probabilities. *Journal of the Construction Division, American Society of Civil Engineers*, vol. 93, no. C01, Proc paper 5159, Mar., pp. 75-107.
- Green, S.D.** (1986) The unbalancing of tenders. MSc dissertation, Department of Building, Heriot-Watt University.
- Green, S.D.** (1989) Tendering: optimisation and rationality. *Construction Management and Economics*, vol. 7, pp. 53-63.
- Kaka, A.P. and Price, A.D.F.** (1991) Net cashflow models: Are they reliable? *Construction Management and Economics*, vol. 9, no. 3, pp. 291-308.

- Kenley, R.** (2003) *Financing Construction: Cash flows and cash farming*. London, Spon Press.
- McCaffer, R.** (1979). Cash flow forecasting. *Quantity Surveying*, August, pp.22-26.
- Skitmore, M.** (2002). Predicting the probability of winning sealed bid auctions: a comparison of models. *Journal of the Operational Research Society*, vol. 53, pp. 47-56.
- Stark, R.M.** (1968) Unbalanced bidding models – theory. *American Society of Civil Engineers, Journal of the Construction Division*, vol. 94, no. C02, pp. 197-209.
- Stark, R.M.** (1972) Unbalancing of tenders. *Proceedings of the Institute of Civil Engineers*, Technical Note 59, vol. 51, pp. 391-392.
- Stark, R.M.** (1974) Unbalanced highway contract tendering. *Operational Research Quarterly*, vol. 25, no.3, pp. 373-388.
- Stark, R.M.** and Rothkopt, M.H. (1979) Competitive bidding: a comprehensive bibliography. *Operations Research*, vol.27, no.2, March-April, pp. 364-390.
- Teicholz, P.M. and Ashley, D.B.** (1978) Optimal bid prices for unit price contract. *American Society of Civil Engineers, Journal of the Construction Division*, vol. 104, no. 1, March, pp. 57-67.
- Tong, Y. and Lu, Y.** (1992) Unbalanced bidding on contracts with variation trends in client-provided quantities. *Construction Management and Economics*, vol. 10, pp. 69-80.

3. OTHER UNBALANCED BIDDING MODELS

3.1 INTRODUCTION

The previous chapter provided some background and this chapter will review the history of unbalanced bidding and the various models for unbalanced bidding that scientists have formulated. It will be shown that the science is still limited by some critical issues and that this has not yet reached the stage where it has gained traction. It has, surprisingly, not yet attracted the same academic interest as ‘bidding models’ nor has any of the research as yet given rise to being applied in practice. Nevertheless, unbalanced bidding is widely practiced using methods that are less scientific.

Unbalanced bidding models have attracted relatively little research compared to the popularity of research on what are commonly known simply as ‘bidding models’. The latter field (having the objective of determining the optimum bid price in a closed tender situation) has attracted over a thousand academic papers. Stark and Rothkopt (1979) had identified over 500 titles on this topic almost 30 years ago.

The research on bidding models is thought to have started with Friedman (1956). Gates (1959, 1967) then contributed an alternative mathematical approach and at the same time he identified the concept of unbalanced bidding models. He suggested that there was most probably no more significant bidding strategy than unbalanced bidding although he felt that this practice entailed little prospect for benefiting from mathematical sophistication.

In practice, over the past 50 years since bidding models and unbalanced bidding models were first identified, the former area could be said to have drawn considerable interest from academics but far less interest from practitioners, and yet one could say the opposite of unbalanced bidding. Many researchers have noted (see, for example, McCaffer, 1979; Green, 1986; Kaka and Price, 1991; and Kenley, 2003) that some rudimentary form of

unbalanced bidding is commonplace in practice. By contrast, however, there has been comparatively little research done in this field.

Gates' initial publication served largely to merely identify the practice. Stark (1968, 1972, 1974) then approached the problem from a mathematical angle. He advocated the use of linear programming which then became a popular standard for much of the subsequent research.

Currently, it is believed that there are three different, complementary approaches to unbalanced bidding (see Cattell, 1984, 1987; Green, 1989; Cattell *et al.*, 2004) namely, front-end loading, individual rate loading, and back-end loading. Very little research has managed to combine all three so as to accomplish what is now believed to be a comprehensive approach. Stark's efforts, for instance, took account only of the combined benefits of two of these (namely, front-end loading and individual rate loading).

Ashley and Teicholz's research (1977), which was subsequent to Stark's (1968, 1972, 1974), nevertheless did not build upon this, but rather, seemingly independently, proposed a simple linear unbalancing model for the sole purpose of front-end loading. In this initial research of theirs, they recognised that some benefit could be derived from individual rate loading but they concluded that it appeared too difficult to systematically model. Teicholz and Ashley (1978) then went on to enhance their earlier efforts in combination with using the contribution of Stark (1974). They proposed a more sophisticated 'optimal model' (resembling that of Gates', 1967) which, despite their initial reservations, managed to expand on their initial effort to now include individual rate loading.

Diekmann *et al.* (1982) took Stark's (1968 and 1974) original deterministic model and, without reference to the works of Ashley and Teicholz (1977), added a probabilistic formulation, using quadratic programming, to take account of risk. Their definition of risk was, however, very limited and they ignored many of the risk factors. They also limited themselves only to the development of a front-end loading model.

Diekmann *et al.*'s (1982) efforts however remain the most significant to-date. Subsequent research by Tong and Lu (1992), which focused solely on individual rate loading, failed to make any advance beyond Stark's initial efforts.

3.2 GATES' STRATEGY

It is believed that Marvin Gates (1959, 1967) was the first to comment on the practice of unbalanced bidding. Gates' entry into this arena has largely been deemed significant because he proposed an alternative to Friedman's (1956) (balanced) bidding model, subsequently coming to criticize the latter (when he became aware of it) for what he considered to be an incorrect use of mathematics (Gates, 1970). Friedman's model pioneered the concept that a contractor might apply mathematics for the purpose of determining the probability of winning a closed tender bid. This mathematical model (being a 'balanced bidding model' by definition) entails analysis of a contractor's competitive environment on the basis that (a) ideally, the contractor has knowledge of who the competitors are, or (b) that it is only known how many competitors there are, or (c) worst of all, that it is not even known how many competitors there are. Friedman's model provides a basis by which a contractor can utilize this analysis by which to determine an optimum bid.

Gates' efforts were largely focused on proposing an alternative mathematical formula by which to determine this probability of winning. The Gates versus Friedman debate has raged on ever since, with many dozens of researchers falling largely into one or other of these two camps (see, for example, Skitmore, 2002). Most bidding models are derived from either Friedman's or Gates' original models. Abdel-Razek (1987) is one of those who have provided a synopsis of the early stages of this conceptual 'battle'; Crowley (2000) provides a more recent assessment, and Skitmore (2002) provides a quantitative comparison.

Whilst the principal area of interest with regards to Gates' research has been with respect to the method of determination of the probability of winning (and hence the determination of an optimum project mark-up), another proposal of his went less well noticed. He is the first researcher to have commented upon the concept of unbalanced

bidding (Gates, 1959). In the process he came to suggest that unbalanced bidding had more to offer a contractor in the short-term as a strategy than any other bidding strategy (Gates, 1959).

He proposed a simple method that addressed a contractor's need for an accelerated cashflow as well as a means to benefit from loading the prices of items whose item quantities are anticipated to increase as a result of a likely variation order. His approach largely steered clear of any sophisticated mathematics and he went so far as to comment that he felt that unbalanced bidding, at least in so far as the manner in which he advocated it, was the least mathematically involved of all the bidding strategies that he was then proposing.

Gates (1967) used the following simple 4-item project to illustrate this strategy:

Table 3.1 Gates' example of a balanced bid (Gates, 1967)

Item No.	Description	Unit	Quantity	Unit Bid, in dollars	Amount, in dollars
1	Clearing	Lump sum	As necessary	50,000.00	50,000
2	Earth excavation	Cubic yards	50,000	1.50	75,000
3	Rock excavation	Cubic yards	25,000	3.00	75,000
4	Cleaning up	Lump sum	As necessary	50,000.00	50,000
Total bid					250,000

Gates proposed that whilst Table 3.1 above illustrates the item breakdown of a \$250,000 project, as it might be without any loading, Table 3.2 below illustrates how it might appear if its items were given the treatment of front-end loading.

Table 3.2 Gates' example of front-end loading (Gates, 1967)

Item No.	Description	Unit	Quantity	Unit Bid, in dollars	Amount, in dollars
1	Clearing	Lump sum	As necessary	90,000.00	90,000
2	Earth excavation	Cubic yards	50,000	1.50	75,000
3	Rock excavation	Cubic yards	25,000	3.00	75,000
4	Cleaning up	Lump sum	As necessary	10,000.00	10,000
Total bid					250,000

Gates hence showed how this project could be made to generate an additional \$40,000 early on in the project which is clearly of value to the contractor to finance his operations.

Gates then went on to suggest that another method of unbalanced bidding was to exploit obvious errors in the project's initial item quantities. He referred to this technique as "unclassifying". He illustrated this with the example of a contractor, having done a field investigation, "having reason to believe" that instead of 50,000 cubic yards of earth excavation, there is more likely to be 70,000 cubic yards; and instead of 25,000 cubic yards of rock excavation, there is more likely to be only 5,000 cubic yards. He suggested that this contractor would benefit if he were to load up the price of earth excavation, and load down the price of rock excavation. He illustrated this with the following example, shown in Table 3.3 below:

Table 3.3 Gates' example of a bid incorporating quantity error exploitation (Gates, 1967)

Item No.	Description	Unit	Quantity	Unit Bid, in dollars	Amount, in dollars
1	Clearing	Lump sum	As necessary	50,000.00	50,000
2	Earth excavation	Cubic yards	50,000	1.60	80,000
3	Rock excavation	Cubic yards	25,000	1.60	40,000
4	Cleaning up	Lump sum	As necessary	50,000.00	50,000
Total bid					250,000

The rate for excavation of \$1.60 was arrived at solely for reason that he proposed that custom dictated that the rate for rock excavation could not be less than the rate for earth excavation. The rate was calculated as the weighted average based on the contractor's "best estimate" of the final quantities of the two items of excavation, as follows:

$$\frac{(70,000 \times \$1.50) + (5,000 \times \$3.00)}{75,000} = \$1.60 \text{ per cu yard}$$

Gates' example drew comparison with the (unaltered) balanced bid, as shown in Table 3.4 below:

Table 3.4 Gates' example of payment on the basis of a balanced bid (Gates, 1967)

Item No.	Description	Unit	Quantity	Unit Bid, in dollars	Amount, in dollars
1	Clearing	Lump sum	As necessary	50,000.00	50,000
2	Earth excavation	Cubic yards	70,000	1.50	105,000
3	Rock excavation	Cubic yards	5,000	3.00	15,000
4	Cleaning up	Lump sum	As necessary	50,000.00	50,000
Total bid					220,000

Gates then demonstrated the obvious benefits of his proposed “classified bid”, depicted in Table 3.5 below:

Table 3.5 Gates' example of payment on the basis of a classified bid (Gates, 1967)

Item No.	Description	Unit	Quantity	Unit Bid, in dollars	Amount, in dollars
1	Clearing	Lump sum	As necessary	50,000.00	50,000
2	Earth excavation	Cubic yards	70,000	1.60	112,000
3	Rock excavation	Cubic yards	5,000	1.60	8,000
4	Cleaning up	Lump sum	As necessary	50,000.00	50,000
Total bid					250,000

Gates thus showed that whilst both his examples of a balanced bid and a classified bid comprised the same tender price of \$250,000, the latter unbalanced bid would generate an additional \$30,000 for the contractor provided that his expectation of the variations to the project's earthworks were to prove correct.

Gates commented that there was, however, considerable risk for a contractor that his predictions as regards variations may not be fulfilled. Notwithstanding this *caveat*, he did not propose any basis by which to measure or address this risk.

Gates' work is believed to be important for reason of having identified unbalanced bidding as a significant strategy. Gates did not, however, succeed with regards to identifying any mathematical techniques by which to accomplish the potential of this strategy. He also did not recognize the potential for the use of sophisticated mathematics for this purpose (Gates, 1967).

3.3 STARK'S MODEL

Stark (1968, 1972, 1974) approached the problem largely as Gates had defined it. Unlike Gates, however, he recognised that there was potential for the application of more sophisticated mathematics and his approach advocated a linear programming solution.

Stark (1968) recognised that there are the following constraints that govern item pricing:

3.3.1 The bid constraint

This constraint simply ensures that all of the items' prices add up to the tender price, which may be stated as follows:

$$TP = \sum_{j=1}^J Q_j P_j \quad (3.1)$$

where

- j = item number
- J = number of items
- Q_j = bill quantity of item j
- P_j = bill price per unit of item j
- TP = tender price

3.3.2 Unit bid constraints

Stark suggested that, for reason of "custom", some item's prices are governed by other item's prices whilst some simply need to be bounded by upper and lower bounds that are seemingly to be arbitrarily decided upon.

These two constraints may be expressed as follows:

$$P_i - P_j \geq 0 \quad (3.2)$$

and

$$P_{jl} \leq P_j \leq P_{ju} \quad (3.3)$$

where i & j = item numbers

P_{jl} & P_{ju} = lower and upper bounds, respectively, for the price of item j

Stark suggested, as an example, that custom dictated that the excavation of hard rock should be priced more than the excavation of soft rock, which in turn should be priced more than the excavation of earth. He made no attempt, however, to provide any

scientific justification for this nature of constraint. Presumably he meant only for this constraint to prevent the situation where otherwise it would draw the attention to the client that the contractor has manipulated his prices.

He also proposed that the upper and lower bounds would be especially useful to limit a contractor's risk where the contractor felt less certain of the final quantity of an item. He did not, however, propose any scientific basis by which to decide these bounds nor did he suggest a scientific basis by which to identify the items to which to apply this technique.

3.3.3 Rate constraints

Stark went on to propose that a contractor may wish to ensure that their anticipated receipts of interim progress payments remain in some proportion to the rate at which the work is being done. He suggested that this constraint be described by way of the following formula:

$$\frac{m}{N} \sum_{j=1}^J Q_j P_j = \alpha \sum_{j=1}^J \sum_{n=1}^m Q_{nj} P_j \quad (3.4)$$

where

- m = a month in the range between 1 and N
- N = the estimated duration of the project, in months
- α = Stark's "constant of proportionality"
- Q_{nj} = the quantity of item j expected to be built in month n

His suggestion was that a contractor should have an "intuitive" feel for deciding an appropriate value of α and that perhaps he should test a range of values within some bounded limits (also to be arbitrarily decided upon). He indicated that a value of 1 should have the effect that interim payments should keep track with the rate of progress, such that when, for example, 20% of the project is completed (measured in terms of value completed, as determined by way of the project's component item prices) then 20% of the overall project's value would be paid to the contractor. Values higher than 1 should have the effect of front-end loading (i.e. that of accelerating the payments) whilst values of less than 1 should give cause for (a basic form of) back-end loading (i.e. that of delaying the payments - not that this was something he advocated).

In effect though, the formulation of this constraint appears fundamentally flawed. If this constraint were to be replicated for all months m , then the only basis by which all these constraints could be satisfied as equalities is in the event of the special case of α being equal to 1 and where the rate of progress, and hence the rate of interim progress payments, of the project, was exactly linear. Besides the impracticality of the latter assumption, this then also negates the effective use of α for the purpose for which Stark intended. Thus, it is believed that Stark failed to formulate this intention correctly. It is also noted that although some subsequent research (see Teicholz and Ashley, 1978 and Diekmann *et al.*, 1982) has made use of other aspects of Stark's linear programme, they have not made use of this constraint.

Stark then expressed his "basic model" as having the purpose of maximising the present value ('PV') described as follows, subject to the constraints listed and described above:

$$PV = \sum_{j=1}^J \sum_{n=1}^N Q_{nj} (1+r)^{-n} P_j \quad (3.5)$$

where r = discount rate of interest, in monthly terms

He described his basic model as being appropriate for those circumstances where the final quantity of each item will exactly match the quantities depicted in the bills of quantities and where the contractor has been exactly able to determine the scheduled rate at which each item will be built. He identified this as a linear programming problem that was capable of being solved within the practical computing constraints of the time.

This basic model thus only addresses the benefit of front-end loading. Stark then developed a further derivative of this model for the purposes of also addressing the benefit of quantity error exploitation and this model (described beneath) entails the maximisation of PV' as follows:

$$PV' = \sum_{j=1}^{J'} \sum_{n=1}^{N'} Q'_{nj} (1+r)^{-n} P_j \quad (3.6)$$

where

J' = the estimated final number of items

N' = the estimated final number of months

Q'_{nj} = the estimated final quantity of item j expected to be built in month n

Stark suggested that there are “some circumstances” in which it will be advantageous to rather optimise the present worth of a project’s future profit (PV''), expressed as follows:

$$PV'' = \sum_{j=1}^J \sum_{n=1}^N Q'_{nj} (1+r)^{-n} (P_j - C_j) \quad (3.7)$$

where C_j = the “known” unit cost of item j

He did not describe which circumstances should give cause to encourage this alternative approach.

This formulation of a project’s profit is simplistic seeing as it does not take account of many factors that determine that a contractor’s cash outflow in respect of any item will seldom, if ever, have a simple and continuous linear relationship with the rate at which the item is built (see, for example, Kaka and Price, 1991; Kenley, 2003).

Stark illustrated this alternative approach by applying to his “basic model” (see equation 3.5 above). It would appear, however, that he intended that it should also be applied to his other model (in which he also incorporates quantity error exploitation – see equation 3.6 above).

Stark’s model was limited to only taking account of the combined benefits of cashflow and quantity estimation errors and it was also limited with respect to its handling of risk. Furthermore, the effect of his model is largely that items are priced at either the upper or lower bounds that the contractor will have imposed as constraints. The model’s ultimate result is thus largely dependent on these arbitrary and subjective inputs, which have to be decided without the benefit of any suggested scientific aides.

Stark did, however, agree with Gates (1967) that there was considerable risk attached to any item price loading strategy. His suggestion was that contractors should test their

pricing models using sensitivity analyses and thereby identify some sense of this risk. This limited approach to risk may be due to his concern about having to design his model so as to be suited to the limited (and expensive) computing resources available to contractors at that stage. His model therefore was purely deterministic in nature.

3.4 ASHLEY AND TEICHOLZ'S MODELS

Whilst Stark's (1968, 1972, 1974) efforts took account of the combined benefits of cashflow and item quantity estimation errors, Ashley and Teicholz's (1977) subsequent research did not build upon this, but rather, seemingly independently, proposed a simple linear unbalancing model for the sole purpose of improving a contractor's cashflow. In this initial research, they recognised that some benefit could be derived from "quantity error exploitation" but they concluded that it appeared too difficult to model systematically.

Interestingly, Ashley and Teicholz's (1977) model was developed for the expressed purposes of determining, *prior* to submitting the bid, the extent and manner to which the cashflow of a project might be manipulated by means of item price loading. They envisaged this information would aid the decision as to whether or not to bid as well as on what amount of bid to submit.

They recommended that, whilst having to cope with the limited amount of information available in the initial stages of a project, a contractor should identify the following three cashflow curves:

3.4.1 Earnings curve -

representing the value of the contractor's "work-in-place", derived from a schedule of activity for the project, prepared by the contractor, priced in terms of the item prices in the bills of quantities.

3.4.2 Payments curve -

derived from the Earnings Curve and adjusted to take account of any retention that may be withheld by the client. This curve will also be 'stepped' (as opposed to being a

continuous ‘smooth’ curve) to take account of the (typically monthly) intervals between client’s payments as well as the delays (between when work is executed, and then measured and then finally paid for). This curve would therefore represent the contractor’s cash inflow.

3.4.3 Cost curve -

representing the contractor’s cash outflow. The cost curve should be derived from the contractor’s estimate of his costs, matched together with his schedule of activity, and taking account of any lead or lag between the timing of the activity and the timing of the expected cash outflow. This curve is also to be derived taking account of the analysis of each item’s cost, comprising the various types of cost components (labour, materials, plant, subcontractors’ charges, and overheads) and how each of these has a different cash outflow lead or lag. The cost curve may bear little or no resemblance to the earnings or payments curves.

Ashley and Teicholz (1977) suggested that the contractor determine his nett cashflow as being the nett difference between the payments curve and the cost curve. They also suggested that these differences per period be discounted back to a present value, using two different discounting rates: one representing the contractor’s cost of borrowings, being applied to any negative nett cashflow amounts; and the other representing his “corporate rate of return” (being larger than his cost of borrowings) being applied to any positive nett cashflow amounts. They concluded that the summation of these periodic (typically monthly) present values would identify the project’s “Nett Present Worth” – useful as a measure of a project prior to a contractor’s commitment to a bid as well as to its constituent item prices.

Ashley and Teicholz (1977) then advocated that a contractor measure the effect of the front-end loading of the item prices using their suggestion of a “linear unbalancing model”, depicted graphically in Figure 3.1 below.

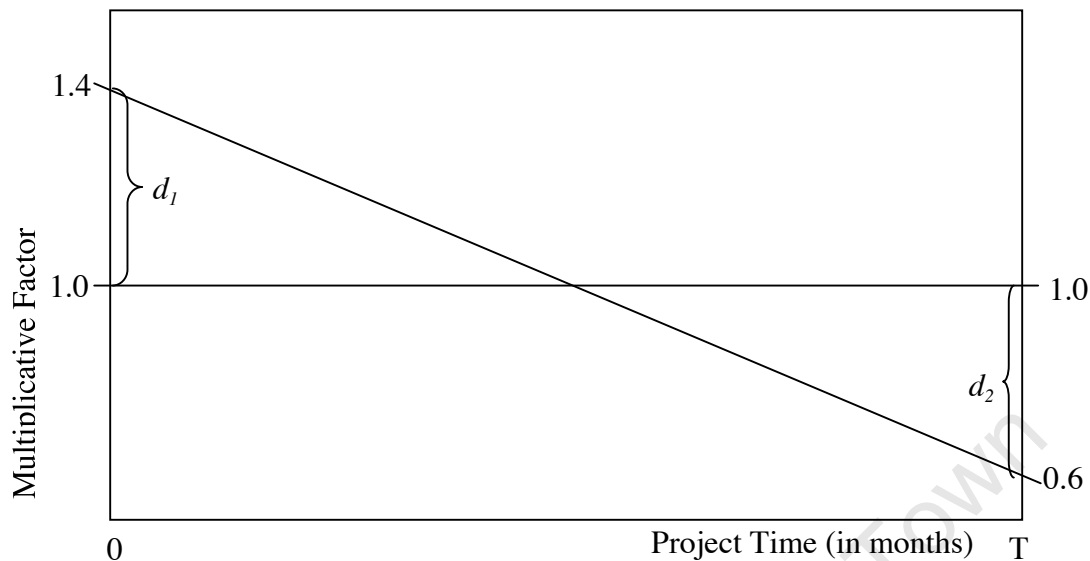


Figure 3.1 - Linear Unbalancing Model (Ashley and Teicholz, 1977)

This linear unbalancing model results in all items scheduled for the start of a project having their prices loaded with the factor d_1 (the “Unbalance Factor”). Subsequent items would be allocated a corresponding lesser factor, derived from the linear equation linking d_1 at the outset of the project with d_2 at the end of the project. Various values of d_1 can be tested so as to determine the effect on the contractor’s cashflow as well as on the project’s “Nett Present Worth”. For each value of d_1 a corresponding, compensatory value of d_2 needs to be determined. Ashley and Teicholz (1977) suggested that d_2 can be found (by means of an iterative search using a computer) such that the overall unbalanced bid retain the same simple cumulative value as the balanced bid, i.e. such that the total “earnings” of the project be kept the same.

Teicholz and Ashley (1978) went on to adopt Stark’s (1974) alternative model with some minor differences. They proposed that this represented a more sophisticated ‘optimal model’ which, by comparison to their initial effort (Ashley and Teicholz, 1977), now included quantity error exploitation.

The differences in their model from Stark’s model may be summarised as follows:

- The optimisation of profit

Although Stark had mentioned this as a possible alternative to the optimisation of a project's revenue, Teicholz and Ashley (1978) give no explanation why they chose only to advocate that a contractor should pursue the maximisation of profit.

- The probability of execution

Teicholz and Ashley (1978) describe the final quantity of an item as the product of two forecasts: the "probability of execution" and the contractor's own estimate of the final quantity.

Their "probability of execution" refers to the contractor's estimate of the chance of any particular item being built. If a contractor believes with certainty that some quantity of a particular item will finally have to be built as part of a project, then they would assign a "probability of execution" for this item of 100%. Similarly, if a contractor were of the opinion that there is only a 50% chance of this item having to be built, then the "probability of execution" would be 50%.

There seems no reason why the latter estimate should not inherently incorporate the former probability. On this basis, the "probability of execution" could be considered a redundant factor. For example, if a contractor were to believe that an item will not finally have to be built, then their estimate of this item's final quantity could simply be expressed as being *nil*.

- The analysis of each item's cost

Stark's model (in which he seeks to optimise a project's profit, as opposed to its revenue) incorporates a simple forecast of each item's cost. This cost is inherently treated as being directly proportional to the item's quantity.

Teicholz and Ashley (1978) incorporate the project's cost estimates somewhat differently. They break the costs down into three categories: fixed, variable and 'interest' (the cost of capital). Their model seeks to optimise the present value of a project's "profit" defined as the revenue less the present value of these three categories of costs.

Moreover, they calculate the present value of the “variable costs” of any item on the basis that it is derived from the final quantity that the contractor estimates as appropriate for this item. Thus their model differs slightly from Stark’s by recognising that some items’ costs incorporate fixed costs, which will not be effected by any variations in the quantities of these items.

- The expressed inclusion of the cost of interest

Stark’s (1968, 1972, 1974) model implicitly incorporates the contractor’s cost of interest with the use of discounted cashflows. Teicholz and Ashley (1978) instead expressly incorporate the cost of interest and yet they advocate, in addition, the use of further discounting.

Their approach therefore has the error of “double counting” the cost of interest, that is, accounting for it twice when only once is correct. It is also flawed by way of calculating the cost of interest by treating each project in isolation. Their calculation only includes the cost of interest in those months when the project’s projected nett cumulative cashflow is negative; alternatively they advocate that there is no cost of interest (that is, they only account for interest paid and do not account for interest earned). This approach fails to recognise that the project being analysed is not one that is in isolation but rather that it inevitable forms part of the contractor’s larger ‘portfolio’ of current projects. The cost of interest to the contractor (incorporating knowledge as to whether the contractor has to borrow money or not) cannot therefore be determined without taking the contractor’s overall cashflow and level of cash resources into account. One cannot consider a single project in isolation and make the assumption that if the project has a negative cashflow that then the contractor’s overall status, inclusive of all his projects, will be such that he will have a negative cashflow. One project’s negative cashflow could, for instance, be offset by another project’s concurrent positive cashflow. Similarly, if a contractor’s status at one stage were such that his overall cashflow were considerably negative, they might be highly dependent, or at very least highly desirous, of as much positive cashflow contribution at that same stage from any new project. A large positive cashflow contribution from the new project would be highly valuable to them relative to a much smaller positive contribution. The former would do very much more to reduce their

overall cost of interest incurred than would the latter. If they were to assess the cost of interest at an isolated project level, it therefore makes more sense that they account just the same for the “cost” of interest regardless of whether, at the project level, the nett cashflow is positive or negative.

- The dropping of Stark’s “Rate Constraints”

Teicholz and Ashley’s (1978) approach does not incorporate the need for Stark’s (1968) “constant of proportionality” (α), which Stark had used as a loading factor to ‘dial up’ the extent of the front-end loading. They have instead been solely reliant on the discounting factor (which is inherent within the method that they used by which to discount the cashflow) by which to achieve the desired degree of pricing preference for early items.

- The “Desirability Index” alternative to the use of Linear Programming

Teicholz and Ashley (1978) suggested that the model could be structured as either a linear programme (as per Stark’s suggestion) or else by using a technique of ranking the items by way of determining a “desirability index” in respect of each item.

They recognised that the effects of their linear programming model (as with Stark’s) gave cause for all items, barring one, to be priced at either of the maximum or minimum pricing limits assigned to each item. Only one item would need to be assigned a price that is not its assigned maximum or minimum limit and this item serves to ensure that the summation of the priced component items will equate with the overall bid price.

Teicholz and Ashley (1978) therefore identified that there was no need for linear programming if a contractor could just rank all of a project’s items in terms of their “desirability”, in other words, in order of their priority status for being awarded or allocated their maximum-limit price. To start with, this approach allocates all items with their minimum-limit price. Next, they calculate the total for all of the items priced accordingly (using the original item quantities in the bills of quantities). If one then deducts this amount from the overall bid amount, one is left with an amount which can serve to be allocated to those items of greatest “desirability”. Thus, starting at the item ranked as having the highest Desirability Index, one allocates this item with its maximum

limit price. One then needs to make the corresponding adjustment to what becomes the remaining balance which is still left to be allocated. Moving down and repeating this process to the next ranked item, and then the next, as the balance reduces one eventually will come to the one item that is to be assigned neither its maximum nor its minimum limit price. This item's price can then be calculated by using up the remaining balance, and thus the problem (as they defined it) is solved.

Their Desirability Index (DI_j) for each item is calculated as follows:

$$DI_j = \frac{P_j \sum_{n=1}^N Q'_{nj} (1+r)^{-n}}{Q_j} \quad (3.8)$$

Their “probability of execution” factor has been left out when translating their formula here for the reason (also explained above) that, in order to be consistent with the formulae throughout this thesis, it is considered unnecessary to have both this probability of execution as well as the estimate of the final quantity. Both can be described by way of the latter variable alone.

Notice that this simplified technique of theirs no longer incorporates estimates of each item's cost. Each item's price (barring one) is effectively being decided as either the minimum or maximum pricing limit influenced by the following three factors:

- the extent to which the item's quantity is expected to be varied,
- the timing of the item in terms of the project's schedule, and
- the discounting rate.

This approach inherently overcomes the “double counting” of the cost of interest discussed above. In effect, this alternative technique of calculation results in a new model that is substantially different in concept from their linear programming (LP) model. *Inter alia*, this new model of theirs seeks to maximise the *revenue* from a project instead of its profit.

This alternative technique also overcomes the need for the contractor to have to incorporate into the model all of each item's estimated fixed and variable costs. This would appear to be both beneficial and problematic. It is problematic because there is merit in identifying all of the fixed costs in a project - those which will arise regardless of variations in the project's item quantities. As long as the model identifies and monitors these fixed costs (should there be any), the model is able to ensure some protection against the risk that in the event that an item's quantity is reduced a proportion of its fixed costs will not have been compensated for. By not monitoring any item's costs, this alternative method of computation loses this advantage.

Besides this advantage, there is no other cause why they should need to incorporate any costs into their LP model. There is no logical reason why a contractor's practice of item pricing should be influenced by knowledge of the individual item costs. By implication, it is also of no benefit to the function of item pricing that the contractor should need to have knowledge of the *timing* of these costs. The contractor's costs, as well as the timing of these costs, will arise regardless of the item price combination chosen by the contractor. There is no causal-effect relationship between the benefits from item pricing, and any individual item's cost. Instead, item pricing gives cause for a substantial influence on a contractor's *revenues*, and also of the *timing* of these revenues.

It follows that any effort to incorporate costs, as well as the timing of these costs, in any item price loading model is a wasted effort. The only apparent reason for Teicholz and Ashley (1978) to have incorporated item costs (and their timing) in their model is their express need to estimate the cost of interest. Their model necessitated estimates of each item's cost of interest (and hence, by summation, they would also have estimated the project's total cost of interest). It has, however, been shown above (see the earlier discussion on "double counting" of interest) that, provided that it is to be considered as an item pricing model, their model had, in fact, no logical need for these cost of interest estimates.

Ashley and Teicholz's (1977) original (cashflow planning) model was, however, presented so as to have a dual purpose: it was firstly said to be intended as a device to

determine the attractiveness of a project to a contractor and hence to aid his decision as to whether or not to bid, and how much to bid. It was also said to be a device to decide appropriate item prices. Given that this model was therefore not solely intended as an item pricing model, there was originally justified cause to have incorporated consideration of each item's cost. In terms of Ashley and Teicholz's (1977) intention of being able to assess a project's 'attractiveness', this requires that the contractor be able to quantify its present worth. This, in turn, requires that the contractor be able to forecast a project's net cashflow, and hence there was the need to incorporate the estimates of each item's cost (and also their timing).

However, Teicholz and Ashley's (1978) subsequent model(s) moved away from being intended to serve this broader purpose. These subsequent models were designed with the express sole purpose of identifying what was intended to be 'optimal' item prices. When considering this transition in their intent of purpose, it can be argued that their LP model no longer had need to incorporate the elaborate analysis of each item's cost in the manner in which it had previously done. Their simplified (desirability-index) alternative model (to their LP model) appears to recognize that this observation is true, notwithstanding that they presented it merely as an alternative, quicker, easier method of calculation, and not as an alternative new model.

Overall, Teicholz and Ashley can be described as having largely abandoned their prior "linear unbalancing model" (Ashley and Teicholz, 1977) and instead having adopted Stark's (1968, 1972, 1974) model. Despite having improved and replaced their earlier model, largely rendering their previous efforts as obsolete, it is their earliest model that is still widely used in recent research (see, for instance, Kenley, 2003).

In summary, although the work of Teicholz and Ashley had the superficial appearance of being substantially original and a significant contribution to this field of knowledge, the effective significance of their work could rather be described as having only identified a quicker and simpler technique of calculation (that can serve as an alternative to the need for linear programming) for the application of Stark's model. It thus did not expand much on what was originally proposed by Stark.

3.5 DIEKMANN, MAYER AND STARK'S MODEL

Diekmann *et al.* (1982) took Stark's (1968, 1972, 1974) original deterministic model and added a probabilistic formulation to take account of risk. All risks, however, were not considered and the only risk to have been incorporated was the risk that the final item quantities may be different from what the contractor initially anticipated. They ignored all other risks (including the risk of variances in all of the variables used in their model), as if they did not exist. What is, however, of valuable significance as the contribution from this research is that their model facilitated that a contractor could utilize item price loading for the purposes of not only maximizing their profit but also of controlling their risk (albeit in the limited manner that they defined it). They thus accomplished the paradigm leap from all previous research (although they made no reference to the works of Ashley and Teicholz (1977) or Teicholz and Ashley (1978)) that had, up until then, simply identified that the practice of item price loading *was* a risk. They recognized that not only could unbalanced bidding contribute to increased risk, but that it also could be used to manage and *reduce* risk. Their model provided the framework for a technique that provided a quantifiable means by which the practice of item price loading could be pursued whilst "balancing" (*their term*) profitability against the measurement and manipulation of a project's risk (no matter that their definition of what constituted risk was limited).

They ignored all risks other than the risk of a variation in item quantities and also only loaded prices in the pursuit of the benefits of cashflow and quantity error exploitation. They therefore ignored the pursuit of increased escalation by way of back-end loading. Their model was also limited by their assumption that the cash inflows and outflows for any item arise simply, without any lead or lag, at the time at which the item is built. Their model thus incorporated the costs of each item but did so without recognizing that the timing of the cash outflows for different items will differ substantially depending on the nature of the item. It can be argued (see the critique of Teicholz and Ashley's model above) that there is no need for this nature of item pricing model to have to incorporate an item's estimated costs nor by implication, the timing of these costs. Nevertheless, Diekmann *et al.*'s (1982) model (also described in Mayer and Diekmann, 1982) does so

although it does so in a manner that may be regarded as simplistic. It does not take account that different items may have substantially different lead / lag times regarding their associated cash outflows. These differences in timing are caused by the various items being composed of different proportions of the constituent types of costs. Some items, for instance, comprise mostly labour costs whilst others may comprise mostly the costs of materials. To forecast any item's cash outflow should require that one give consideration to the item's cost break-down into its constituent components which may be of many different types (labour, materials, sub-contractors' charges, etc.), each with substantially different delays / advances between when the item is built and when each of these costs are having to be paid for.

Whilst their model is derived from Stark's (1968, 1972, 1974) model, they did not implement Stark's "constant of proportionality" (α) constraint. Incidentally, this constraint was also ignored in the subsequent work done by Ashley and Teicholz (1977).

Their model as regards the expected value of PV is given below. Their equation has been translated (as with all the equations throughout this thesis) to a common format in which the same symbols are used throughout.

$$E(PV) = \sum_{j=1}^J A_j \bar{Q}_j (P_j - C_j) - F_0 \quad (3.9)$$

$$\text{where } A_j = \sum_{n=1}^N (1+r)^{-n} Q_{nj} / Q_j \quad (3.10)$$

\bar{Q}_j = the mean or expected value of Q_j

and F_0 = the present value of the fixed costs of the project.

Thus, Diekmann *et al.* (1982) have presented this model as the probabilistic equivalent of Stark's (1974) deterministic model. They go further to show that, with the assumption that the item quantities are normally distributed, the formulation of the mean and variance of the expected profit are as follows:

$$\mu = \sum_{j=1}^J A_j \bar{Q}_j (P_j^\circ - C_j) - F_0 \quad (3.11)$$

$$\sigma^2 = \sum_{j=1}^J A_j^2 \sigma_j^2 (P_j^\circ - C_j) - F_0 \quad (3.12)$$

on the assumption that the covariances between the items are zero

where σ_j^2 is the variance of Q_j

and P_j° is the present value of P_j

If, on the other hand, the variances in the quantities of the items are not mutually independent, then the variance is given as follows:

$$\sigma^2 = \sum_{j=1}^J \sum_{i=1}^J A_i A_j \sigma_{ij} (P_j^\circ - C_j) (P_i^\circ - C_i) \quad (3.13)$$

where σ_{ij} represents the covariance between the variables Q_i and Q_j for $i \neq j$ or the variance of Q_i when $i = j$.

Diekmann *et al.* (1982) chose to measure risk by way of assessing the probability that the present value of a project's projected profit not be less than some chosen value V . They go on to show that, for any given value V by way of this definition, the higher the mean μ the lower the risk, and the lower the variance σ^2 the lower the risk.

On this basis, they choose to combine these factors in one measure of a project's risk by way of the following objective function (which a contractor is obviously wishing to have maximized):

$$R_c = \mu - k\sigma^2 \quad (3.14)$$

in which μ is defined by way of equation 3.11 above and σ^2 is defined by way of either 3.12 or 3.13, as appropriate.

This model can be solved as a quadratic programme seeing as it comprises an objective function that is of a quadratic format, which is to be solved subject to linear constraints. The linear constraints they use are the same as those identified by Stark (1968, 1972, 1974), namely those given as equations 3.1, 3.2 and 3.3 above.

Diekmann *et al.* (1982) argue that the value V can be intuitively chosen by a contractor to suit his “economic and competitive situation”, i.e. that the chosen level of V will depend on his willingness to accept risk.

The contractor’s choice of constant value k is even more abstract in concept and more likely to be difficult to decide on. The constant k has the effect of shifting emphasis between the mean μ and the variance σ^2 as being the two sources of risk (as they have chosen to define it). High values of k have the effect of giving preference to minimising the project’s projected profit variance, and hence of avoiding the pursuit of quantity error exploitation, unless the contractor is reasonably certain of the anticipated quantity variation. Very low values of k , e.g. values approaching zero (bearing in mind that k is restricted to being a positive value), will give cause for the function in equation 3.11 to effectively resemble the function depicted in equation 3.9 in the sense that little to no regard will be made of any item’s expected variance.

Diekmann *et al.* (1982) appear willing to accept that the contractor’s choice of the values V and k is difficult and likely to be problematic. In essence they suggest that one experiment with different values (with there being little guidance available on this matter, nor that there are any ‘right’ or ‘wrong’ approaches to this) judging to see what effect different scenarios have on the resultant item prices, as well as on the mean and variance of the expected profit.

Diekmann *et al.* (1982) provide an example to illustrate the favourable effect of their model. Unfortunately, in this example they choose to withhold imposing any upper limits to the price of each item even though they recommend that in practice these limits are required. Furthermore, in the manner in which the example is captured as a quadratic programme, it limits each item’s price to reflect a minimum mark-up of *nil*. No

explanation is provided as to why some item's prices are not allowed to be less than the corresponding estimated item cost.

Without the imposition of upper price limits, this example therefore enjoys greater freedom by which to pursue its objective. The result is that it does not illustrate at all the problem that arises when one imposes arbitrarily chosen upper and lower pricing limits for each item. What happens in the event of upper and lower pricing limits is that all items barring one will be assigned a price corresponding to either one of these two limits. Thus, in practice, in accordance with their recommendation (and not in accordance with their example) it can be argued that there is no greater influence on this model than these arbitrarily chosen limits. Nevertheless, as with Ashley and Teicholz (1977), Teicholz and Ashley (1978) and Stark (1968, 1972, 1974), the sophistication of this model does not extend to addressing this critical issue.

In conclusion, one could argue that this model constituted a very significant contribution to the science of unbalanced bidding. It was the first model to address the management of both the profitability to be derived from item price loading as well as the risk. However, although Diekmann *et al.*'s (1982) model is of far greater mathematical sophistication than any model that preceded it, it is believed to be flawed for reason of the following limitations:

- the consideration that a project's risk is solely related to the risk of variation in each item's quantities (with the treatment of all the other variables being in a deterministic way),
- the unnecessary complexity of the incorporation into the model of having to forecast a project's cash outflow (for the purposes of optimizing a project's profit rather than its revenue),
- the contractor having to decide on values (without any scientific aid) for some abstract constants (namely, V and k) which have a critical influence on the model's outcome,

- the effective use of the constant k by which to ‘balance’ the risk vs. return trade-off, where k is a very arbitrary and abstract measure of this critical decision,
- the use of two sets of constraints whereby each item’s price is simplistically a function of other items’ prices, and also that each item’s price is constrained by simplistic upper and lower limits (thus it suffers from the same problems as Ashley and Teicholz’s (1977) model as well as Stark’s (1968, 1972, 1974) model), and
- the limitation to the pursuit of front-end loading and quantity error exploitation only and hence the failure to recognize the benefits of back-end loading.

3.6 TONG AND LU’S MODEL

Tong and Lu (1992) developed a method that was focused solely on optimizing the advantage of what they called ‘error exploitation unbalancing’ (referred to by Green (1986) as ‘individual rate loading’ and by Cattell (1987) as ‘loading for anticipated quantity variation orders’). In other words, this method ignored the other benefits in the areas of cashflow and escalation.

Tong and Lu (1992) considered two alternative models – with the same intended effect: one using linear programming (LP) and the other using a method they called their “minimum-maximum method”. Their sole motivation and justification for this second method was due to what they considered the impractical scale of the LP model. They were of the opinion that the Simplex Algorithm would not cope with being able to solve an LP model of the size they were envisaging. However, this runs contrary to current popular computerized usage of the Simplex Algorithm (see Williams, 1993) which manages to fairly quickly be able to solve models comprising well over a million constraints (provided that all the variables are continuous and not integral). This ease of solving large LP models, incidentally, contrasts with greater difficulty in solving Mixed Integer Linear Programmes (MILP) (where all the variables are not continuous – some variables, by definition, being integers or binary variables) which take far longer to solve. It is typically with MILP (or ‘MIP’), rather than with pure LP, where the size of the

model is likely to be of practical concern. In particular, the concern is typically with regards to the number of integer and binary variables, rather than the number of continuous variables. In the case of Tong and Lu's (1992) LP model (in which all the variables are, by implication, continuous), it should not present any practical concern as regards being quick enough. With this being the case, there should not be need to have to consider their 'practical' alternative method.

Tong and Lu's (1992) alternative model (their 'minimum-maximum model' that they proposed as quicker and easier than their LP model) resembles the 'Desirability Index' model from Teicholz and Ashley (1978). Teicholz and Ashley had likewise proposed their 'Desirability Index' model as their alternative to their LP model. Tong and Lu (1992), however, make no reference to the work of Teicholz and Ashley (1978).

Their 'minimum-maximum model' therefore did not present any advance on the model of Teicholz and Ashley (1978). Their alternative LP model, similarly did not present any practical advance on the modeling technique proposed by Stark.

3.7 AFSHAR AND AMIRI'S MODEL

Recent research by Afshar and Amiri (2008) advocates using a Fuzzy Linear Programming (FLP) approach. This research suggests that all previous unbalanced bidding models have "*disregarded risk and uncertainty*" (Afshar and Amiri, 2008: 58). This assertion obviously fails to adequately acknowledge the efforts of Cattell (1984, 1987), and Cattell *et al.* (2004). Cattell *et al.* (2007) even went so far as to have advocated the use of fuzzy constraints on a linear programming model, but this is also not acknowledged.

Afshar and Amiri's (2008) model is built on the assumption that Stark's (1974) LP model serves to best define the status-quo as regards the deterministic modelling of unbalanced bidding. This assumption is obviously flawed, since Stark's (1968, 1972, 1974) model was an early effort. Nevertheless, Afshar and Amiri's (2008) model amounts to an FLP adaptation of Stark's (1968, 1972, 1974) LP model.

Afshar and Amiri (2008) fail to recognise that many aspects of unbalanced bidding models entail uncertainty. They have focused in on the fuzzy characteristics of the upper and lower price limits and quantity variations and have ignored the risks inherent in all of the other uncertain assumptions that are being made. For example, they have ignored that there is considerable risk that these models can be misleading due to the uncertainty in the discounting rates that are used in the Present Value calculations. There is also usually considerable uncertainty in the scheduling of the project, particularly at the early stage at which the item pricing is being decided.

In summary, this research is believed to represent a move in the right direction. It has taken a proposal of the use of fuzzy modelling (made in Cattell *et al.*, 2007) and has applied it. Unfortunately, this application has ignored most of the uncertainties involved in the domain and the tests that are provided are inconclusive as regards the merits of FLP.

3.8 CHRISTODOULOU'S MODEL

Recent research by Christodoulou (2008) has proposed minimizing the *financial entropy* (disorder) of a project as a measure of some aspects of the risks involved. This research does not, however, provide a comprehensive measure of all of the risks involved.

Christodoulou's (2008) approach has been to adopt the model provided by Cattell *et al.* (2007) (as provided in Chapter 4) and to build upon this. He goes on to evaluate *entropy* (H_x) by way of the formulation provided in equation 3.14.

$$H_x = p_x \ln \left(\frac{1}{p_x} \right) \quad (3.14)$$

given that p_x is the '*probability distribution*' of variable x , that is, a measure of its uncertainty / risk.

This research constitutes a novel approach but suffers from the following flaws:

- the model requires the imposition of fixed upper and lower limits on item mark-ups, without providing any explanation of how these are to be decided and without recognition of the importance of these as constraints,
- it treats the item costs as deterministic variables, without recognizing the considerable uncertainty that is inherent in estimating these (see Beeston, 1975),
- it fails to recognize that the summation of the item prices must exactly equal the project's overall tender price, and the model presents the possibility that the item prices may add up to less than the tender price, which would violate the typical unit price contract,
- it assumes a normal distribution of item prices, without apparent cause,
- it assumes that these normally distributed prices are centered around prices derived from an equal mark-up being applied to all of a project's items – rather than with reference to any market's perception of these prices,
- it assumes a 10% standard deviation for these normal distributions, without apparent cause, other than that this 'small variance' will have the effect of 'safeguarding' against 'unreasonably high item mark-ups' being 'disallowed',
- it assumes an interest rate, to be used for discounting, known with complete confidence,
- it fails to recognize the prospective variation in item quantities and, therefore, does not facilitate any individual rate exploitation,
- it fails to recognize escalation and therefore, does not counter-balance any front-end loading with any back-end loading,
- it presents no solution as regards whether to give more emphasis to the pursuit of more profit or less *entropy*, and
- although the model has been 'tested', this fails to prove convincing.

3.9 CONCLUSION

Item pricing models have commonly been designed as linear programming models. Linear programming has the restriction that a model may only have one objective function. This restriction appears to have led to the practice in which early item pricing models sought only to maximize either a project's profit or else its revenue. These models consequently did not incorporate any assessment of a contractor's risk.

Diekmann *et al.* (1982) chose instead to use quadratic programming and whilst this technique again has the restriction of only one objective function, they proposed an abstract way (facilitated by being able to use a quadratic equation as the objective function) of pursuing a maximization of the expected profit together with a minimization of the risk. They effectively came to combine these two objectives with the use of a constant (k) for which the contractor has to arbitrarily choose a value. Different values of k have the effect of shifting the emphasis of the objective from the maximization of profit to the minimization of risk.

The only other method that has been found to be employed in all of these models by which to control "the risk" is by means of the use of constraints by which to impose arbitrarily chosen (upper and lower) limits on each item's price. These limits, rather than any other factor, contribute the greatest effect to both the extent to which a project's profit can be maximized as well as the extent to which a project's risk can be minimized. However, despite the significance of these limits, it is commonly recommended that they should be decided upon without any scientific or mathematical aid. The use of advanced mathematical programming to refine other aspects of item price loading therefore appears somewhat superfluous as long as the most influential factor, is by comparison, left to be handled relatively crudely.

In effect, the only role that the mathematical techniques have to play is to determine the sequence by which to prioritise which items are to be allocated their upper limit price. This sequence thus identifies the one remaining item that will fall between the set of items given their upper limit prices and the set of items given their lower limit prices.

This one remaining item thus serves to satisfy the constraint by which the unbalanced bid is, in summation, equal to the already-determined tender price.

Another critical and yet common shortfall of the above-described models relates to their definition of risk. Even in the case of Diekmann *et al.*'s (1982) model, the only risk factor that was considered and modelled was the risk that the final quantity for some items may be different from that which the contractor estimates. No consideration is made of the risks that are generated from 'mis-estimates' of the most appropriate discounting rate, of cost estimates, of changes to the project's schedule, nor any of the other factors that these models simply treat as worthy of a deterministic usage.

Thus, the state-of-the-art as regards unbalanced bidding models suffer from the following shortcomings:

- they are limited in their recognition and management of what constitutes risk,
- none provide a comprehensive (and hence balanced) technique by which to pursue all three of the recognised methods of item price loading (namely, front-end loading, individual rate loading and back-end loading),
- only Diekmann *et al.*'s (1982) model provides any means by which to quantifiably address the risk vs. return trade-off, but this technique provides a rather abstract device that is not capable of simple, intuitive use, and
- none provide any device by which a contractor can assess the merits and demerits (in terms of both profit and risk) of any adjustments to the upper and lower pricing limits of each individual item – and yet it will be shown that this set of constraints, rather than any other factor, has the greatest influence on the effects of unbalanced bidding. All of the models described above also share in common that they recommend that these limits be arbitrarily decided upon (and regarded thereafter as fixed). In reality, it is highly unlikely that any item should have a maximum or minimum price beyond which a contractor would never submit a price, regardless of the benefits of doing so in terms of increased profit and / or

decreased risk. In practical terms, these limits may be regarded as more fuzzy than fixed.

There was a lapse of 15 years (from 1992 to 2007) in which little was proposed in the field by way of new models. Cattell *et al.* (2007) provided a synopsis of the prior research and identified the shortfalls, particularly as regards the modelling of the risks involved. More recent research (see Afshar and Amiri, 2008; Christodoulou, 2008) has addressed this and become more focussed on the reduction of risk rather than the pursuit of profit. Thus far, no convincing models have emerged that are able to withstand vigorous scrutiny.

The chapters that follow describe the new model that has been formulated. Chapter 4 is focussed on the modelling of a contractor's benefits that they will derive from a project.

REFERENCES

- Afshar, A. and Amiri, H.** (2008) A fuzzy-based model for unbalanced bidding in construction. First International Conference on Construction in Developing Countries, August 4-5, Karachi, Pakistan. Available at <http://www.neduet.edu.pk/Civil/ICCIDC-I/Complete%20Proceedings.rar#page=68> [accessed 8th April, 2009]
- Abdel-Razek, R.H.** (1987) Computerised analyses of estimating inaccuracy and tender variability: causes; evaluation and consequences. Unpublished PhD thesis, Loughborough University of Technology.
- Ashley, D.B. and Teicholz, P.M.** (1977) Pre-estimate cash flow analysis. *Journal of the Construction Division, American Society of Civil Engineers*, Proc. Paper 13213, vol. 103 no. C03, pp. 369-379.
- Beeston, D.T.** (1975) 'One statistician's view of estimating.' *Chartered Surveyor, Building and Quantity Surveying Quarterly*, vol. 12, no. 4, pp. 49-54.
- Cattell, D.W.** (1984) A model for item price loading by building contractors. Unpublished paper, Department of Quantity Surveying, University of the Witwatersrand, Johannesburg.
- Cattell, D.W.** (1987). Item price loading. *PACE'87 Progress in Architecture, Construction and Engineering proceedings*, Johannesburg, July, Vol.II, Session 3, pp. 1-20.
- Cattell, D.W., Bowen, P.A. and Kaka, A.P.** (2004) A model to distribute mark-up amongst quotation component items: an outline, *Proceedings of the 2nd Postgraduate Conference on Construction Industry Development*, Cape Town, Oct, CIDB, pp. 154-165.
- Cattell, D.W., Bowen, P.A. and Kaka, A.P.** (2007) Review of unbalanced bidding models in construction, *American Society of Civil Engineers, Journal of Construction Engineering and Management*, vol. 133, no. 8, Aug, pp. 562-573.

- Christodoulou, S.E.** (2008) A bid-unbalancing method for lowering a contractor's financial risk. *Construction Management and Economics*, vol. 26, pp.1291-1302.
- Crowley, L.G.** (2000) Friedman and Gates – another look. *Journal of Construction Engineering and Management*, vol. 126, no. 4, July, pp. 306-312.
- Diekmann, J.E., Mayer, R.H. Jr. and Stark, R.M.** (1982) Coping with uncertainty in unit price contracting, *Journal of the Construction Division, American Society of Civil Engineers*, vol. 108, no. C03, Sep, pp. 379-389.
- Friedman, L.** (1956) A competitive bidding strategy. *Operations Research*, vol. 4, no. 1, Feb., pp. 104-112.
- Gates, M.** (1959) Aspects of competitive bidding. *Annual Report*, Connecticut Society of Civil Engineers.
- Gates, M.** (1967) Bidding strategies and probabilities. *Journal of the Construction Division, American Society of Civil Engineers*, vol. 93, no. C01, Proc paper 5159, Mar., pp. 75-107.
- Gates, M.** (1970) Reply to Morin and Clough (June 1969) *Journal of the Construction Division, American Society of Civil Engineers*, vol. 96, no. C02, June, pp. 93-97.
- Green, S.D.** (1986) The unbalancing of tenders. MSc dissertation, Department of Building, Heriot-Watt University.
- Green, S.D.** (1989) Tendering: optimisation and rationality. *Construction Management and Economics*, vol. 7, pp. 53-63.
- Kaka, A.P. and Price, A.D.F.** (1991) Net cashflow models: Are they reliable? *Construction Management and Economics*, vol. 9, no. 3, pp. 291-308.
- Kenley, R.** (2003) *Financing Construction: Cash flows and cash farming*. London, Spon Press.
- Mayer, R.H. and Diekmann, R.E.** (1982) Enhancing profitability for unit price contracts. *AACE Transactions*, American Association of Cost Engineers, Houston, Texas, USA, pp. M.3.1-M.3.7.
- McCaffer, R.** (1979). Cash flow forecasting. *Quantity Surveying*, Aug, pp.22-26.

- Skitmore, M.** (2002) Predicting the probability of winning sealed bid auctions: a comparison of models. *Journal of the Operational Research Society*, vol. 53, pp. 47-56.
- Stark, R.M.** (1968) Unbalanced bidding models – theory. *Journal of the Construction Division, American Society of Civil Engineers*, vol. 94, no. C02, pp. 197-209.
- Stark, R.M.** (1972) Unbalancing of tenders. *Proceedings of the Institute of Civil Engineers*, Technical Note 59, vol. 51, pp. 391-392.
- Stark, R.M.** (1974) Unbalanced highway contract tendering. *Operational Research Quarterly*, vol. 25, no. 3, pp. 373-388.
- Stark, R.M. and Rothkopt, M.H.** (1979) Competitive bidding: a comprehensive bibliography. *Operations Research*, vol. 27, no. 2, March-April, pp. 364-390.
- Teicholz, P.M. and Ashley, D.B.** (1978) Optimal bid prices for unit price contract. *Journal of the Construction Division, American Society of Civil Engineers*, vol. 104, no. 1, March, pp. 57-67.
- Tong, Y. and Lu, Y.** (1992) Unbalanced bidding on contracts with variation trends in client-provided quantities. *Construction Management and Economics*, vol. 10, pp. 69-80.
- Williams, H.P.** (1993) *Model solving in mathematical programming*, West Sussex: Wiley.

4. MODELLING A PROJECT'S REVENUE

4.1 INTRODUCTION

Chapter 3 has reviewed the research undertaken as regards the development of unbalanced bidding models other than the one that is the subject of this thesis. It highlighted that one weakness of these other models (with the exception of the models proposed by Diekmann *et al.* (1982) and the two recent efforts by Afshar and Amiri (2008) and Christodoulou (2008)) has been their failure to use item price loading as a mechanism by which to measure and control a project's risk.

Diekmann *et al.*'s model pioneered the approach that item price loading is not only capable of creating additional risk for the contractor, but also that it can be used to measure and control, and even reduce, a project's risk. This quadratic programming model came to combine a contractor's objectives of both wishing to maximise his profit and minimise his risk. It did so by way of a single equation, using a constant (k) as the basis of creating a balance between these two objectives.

The model that is being proposed by way of this thesis is similar to Diekmann *et al.*'s model in as much as it also recognises that item price loading can be used to serve both of a contractor's objectives (namely, of maximising profit and minimising risk). However, the new model is also very different (in this regard, besides others) in the manner in which it combines these two objectives. This model comprises an initial modelling of the revenue (or return) from a project quite separately from its modelling of a project's risk. These two sets of results are then combined: in effect implementing a use of modern portfolio theory (Chapter 6) and cumulative prospect theory (Chapter 7). Ultimately, the (combined) model identifies the optimum item prices best suited to the contractor's objectives (being both profitability and risk reduction).

On this basis, this chapter will discuss the modelling of a project's 'revenue'. It will be shown that this model accomplishes the same result, in terms of identifying the performance of various item pricing combinations, as it would if it were instead to model a project's 'profit' or 'return'.

This chapter describes 'front-end loading', 'back-end loading', and then 'individual rate loading' (which were all introduced briefly in Chapter 2) before it describes the new composite model that incorporates all three of these. It then discusses the constraints normally applied to models in this domain and describes whether these should again be applied in this instance.

The next chapter (Chapter 5) will discuss the modelling of a project's risk and the chapter thereafter (Chapter 6) will address combining the aspects of return and risk with the use of modern portfolio theory. Chapter 7 will then discuss cumulative prospect theory and how indifference mapping might be used to help identify one *optimum* choice from the range of *efficient* choices that portfolio analysis will have identified. Chapter 8 will show how one combined model, that uses Monte Carlo Simulation, succeeds in moderating the opposing objectives of risk and return. In order to make a logical progression to this point, this chapter will explain the modelling of a project's revenue as if the aspect of risk were not a consideration or an issue that were to be involved.

4.2 THE RANKING OF ITEMS

The objective for any unbalanced bid is to determine the optimum distribution of a project's overall tender price amongst its multitude of component items. To satisfy this objective, one might look to maximise a project's profit assuming that one is able to determine such profit from each possible item price combination. To accomplish this suggests that one must be able to quantify the profit contribution from each item, if assigned a particular price, such that the project's profit will amount to a summation of all of the contributions of profit from each of these items. This thesis advocates that each item's benefit can be determined by way of its profit contribution relative to its price (and this will later be shown to be a linear relationship).

This concept is similar to what Teicholz and Ashley (1978) referred to as each item's 'Desirability Index'. This approach was advocated by Cattell (1987) and is also similar to the approach that Tong and Lu (1992) adopted as regards their maximum-minimum technique. This measure of each item's profit relative to price shall be referred to hereafter as the Profitability Responsiveness Index ('PRI') and the ranking of a project's items in terms of this index, as the Profitability Responsiveness Item Ranking ('PRIR').

Having the PRI facilitates a contractor knowing whether the combined effects of any item's characteristics should rank as being more 'worthy' of being allocated a higher price than other items. This (PRIR) ranking of items should therefore list the items in the sequence that they promise to reward the contractor with added profit, in response to being allocated a higher price. Thus, the item with the highest PRIR is the item that has been identified as most likely to generate the best return for being allocated a unit of currency. If the contractor's objectives were only that of profitability, then it is the one item that a contractor should most wish to price as high as possible. The PRI comprises a summation of similar values that are derived from each of the practices of front-end loading, individual rate loading and back-end loading.

4.3 FRONT-END LOADING

If front-end loading were to be pursued in isolation, one would desire to mark-up as high as possible the items that are scheduled to be built early on in the project. The objective of this practice is for the contractor to generate as much cash as possible, as quickly as possible.

One might make the following assumptions as the basis by which to initially simplify the building of this model:

- that there is no practice of retention,
- that there is no practice of escalation,
- that there is no possibility of any quantity variation,

- that the costs of any item j are incurred simply simultaneously with the receipt of the interim payment for that same month, and,
- that the costs when incurred are exactly as they were estimated and are fixed (in the sense of not being subject to inflation) at the time of the estimate.

Some of these assumptions may seem unreasonable but they are intended to serve two purposes, namely:

- so that the model initially appears as simple as is possible, holding back on some of the complexities so that they are added only later, once the basic model has been formulated, and
- because it will become obvious later that some of the variables in the initial model become irrelevant as later phases of the modelling process are introduced. It is therefore not necessary to have to burden any elegant simplicity of the initial phases of the modelling process with superfluous complexity that will only later be discarded when found to be redundant.

On this basis, considering only the purpose of pursuing front-end loading (i.e. of pursuing the improvement of a project's cashflow) a contractor could determine the present value (PV_j) for any item j (that shall later form the basis of the PRI for that same item) by way of the following equation:

$$PV_j = \sum_{n=1}^N \left(\frac{1}{1+r_j} \right)^n \left[\lambda_{nj} Q_j (P_j - C_j) \right] \quad (4.1)$$

where

- j = item number
- n = month number
- N = duration of project in months
- r_j = discount rate appropriate to the risk of item j
- λ_{nj} = proportion of Q_j to be built in month n
- $\sum_{n=1}^N \lambda_{nj} = 1.0$ for any item j

$$Q_j = \text{bill quantity of item } j$$

$$P_j = \text{bill price per unit of item } j$$

$$C_j = \text{unit cost of item } j$$

Items that are scheduled to be built early on in the project will have high values of λ_{nj} when n is of a low value. For these ('early') items, when n tends to N , λ_{nj} is most probably 0, or at very least of very low positive value (i.e. $\rightarrow 0$). One can easily see from this equation that these early items will generate a higher PV_j than the equivalent item scheduled to be built later in the project. This corresponds to the principle of front-end loading in that it clearly recognises that early items priced highly will generate higher PV_j s than later items priced similarly.

Furthermore, it obviously follows that the higher the discount rate r_j the greater will be the differential between the PV_j generated by early and late items.

4.4 INDIVIDUAL RATE LOADING

As noted previously, the loading of the rates of individual items is a practice otherwise referred to as quantity error exploitation (see Tong and Lu, 1992). This practice amounts to allocating high prices to items whose initial quantities in the bills of quantities are thought likely to be increased, whilst allocating low prices to items whose billed quantities are thought likely to decrease.

This practice arises in those forms of contract where the initial contract quantities are not fixed as final but instead are subject to review and adjustment as the realities of the project unfold. An example of this would be projects let on the basis of provisional bills of quantities. These adjustments are typically necessitated by initial uncertainties in the design, or by the soil and rock conditions on site being found to be different from that which was initially expected.

This practice was first commented on by Gates (1959) and has subsequently been referred to many times (see, for instance, Gates, 1967; Diekmann *et al.*, 1982; Cattell,

1984, 1987; Tong and Lu, 1992) as something that is very common and widespread amongst contractors.

The benefit to a contractor from individual rate loading is derived from their ability to shift their margin onto items where, when the consequently high prices (with high margins built in) are applied to increases in these items' quantities, the contractor enjoys a far greater compensation for his extra work than is reflected in his increased cost of the added work. Furthermore, a contractor can use the opportunity of a prediction that an item's final quantity will be less than its initial quantity in the bills of quantities, by allocating such items a low price. If the prediction is correct, the ultimate reduction in the payment made to the contractor will be less than if they were to have priced such an item any higher.

On the basis of the same assumptions as made above under heading 4.3 (with the obvious exception of quantity variations) and considering only the purpose of pursuing individual rate loading (i.e. the pursuit of the exploitation of any errors or other adjustments in the billed quantities), a contractor could quantify the present value (PV_j) for any item j (that shall later form the basis by which to determine the PRI for that same item) by way of the following equation:

$$PV_j = (Q_j + Q'_j)(P_j - C_j) \quad (4.2)$$

where Q'_j = additional quantity of item j due to variation

Items that have a high Q'_j will generate a high PV_j especially when combined with the allocation of a high price P_j . Thus, any model that has the objective of maximising the summation of the PV_j 's will give cause for high prices to be allocated to those items that have relatively high Q'_j 's. The reverse is true of items that have relatively low (typically negative) Q'_j 's and hence these items create the "funding" differential by which the margin can be shifted from those items of low Q'_j 's to those items of high Q'_j 's.

4.5 BACK-END LOADING

The opportunity with back-end loading is for a contractor to be over-compensated for the inflationary increases in their expenses. This opportunity arises in contracts that incorporate the practice of escalation payments in terms of contract price adjustment provisions. In such situations an estimate is made of the contractor's actual cost of inflation with the objective being that they should be compensated for this added expense. The concept is such that the contractor should neither profit nor make a loss from inflation but rather that any risk that comes from inflation should be passed over from the contractor to the project's developer.

In such contracts, contractors do not simply provide proof of their actual costs of inflation; although historically this was done and was known as the 'proven cost' method of reimbursement. Contractors' initial estimates of their costs constitute confidential information and whilst they are known only to them, it is not appropriate for the developer or their professional agents to have access to this. Neither is it appropriate for developers or their agents to have access to knowledge of the final cost that is incurred by the contractor. Moreover, the sheer volume of paperwork involved would render it impractical to employ a 'proven cost' method of dealing with inflation. This nature of contract therefore incorporates the concept of 'escalation' by which an estimate is made of the actual increase in the contractor's costs. This escalation estimate provides the mechanism by which an amount roughly representing the actual increase in costs can be borne by the developer, rather than the contractor. Published indices, such as those known in South Africa as Haylett Indices (JBCC, 2005), are usually used for this purpose.

The opportunity for back-end loading stems from the fact that the values of escalation is, in such instances, determined from estimates based on the contractor's gross item prices, rather than being based on actual costs. For instance, in South Africa the escalation calculation is done in terms of the "Haylett" contract price adjustment provisions (JBCC, 2005), with a non-adjustable element of 15% i.e. the adjustment factor is 0.85. This implies that it is being assumed that a contractor's cost is 85% of any item's price. Thus, if a contractor's price is high, their assumed cost is also high, regardless of their actual

cost. One simple way for a contractor to practice back-end loading is therefore to apply high prices to items that are scheduled to occur late in the contract. These high prices will give the impression that these items have high costs and therefore they will enjoy high levels of escalation adjustment; provided that the expectation is proven correct as regards the increase in these indices.

Another opportunity to benefit from back-end loading is derived from the system by which items of work are typically categorised into escalation workgroups. This system is structured so that inflation is monitored in each of the workgroups and indices published specific to each of them. This method facilitates that if, for example, the cost of iron-ore 'sky-rockets', that the work that entails the use of this material will be adequately compensated for. The potential for loading arises from the ability for a contractor to make predictions of the escalation rates in each of the workgroups. They could allocate higher prices to those items that fall within the workgroups that are expected to have higher than average rates of escalation.

Both of the above two techniques could be combined. Contractors practicing back-end loading should ideally allocate the highest prices to those items that are scheduled for late in the contract and also which are categorised as falling into the workgroups that are expected to have the highest rates of escalation. The 'funding' for these high prices obviously needs to be sourced from the use elsewhere of relatively low prices, the lowest of which should be allocated to the items that are scheduled for completion early in the project and which furthermore are categorised into the workgroups that have the lowest expected rates of escalation.

The following formulation describes the basis by which an item's present value PV_j can be quantified:

$$PV_j = \sum_{n=1}^N [(\lambda_{nj} (Q_j + Q'_j)) [\Upsilon_{nj} f P_j - C''_{nj}]] \quad (4.3)$$

where

$$\Upsilon_{nj} = \text{adjustment for escalation} = \frac{\text{index}_n - \text{index}_0}{\text{index}_0}$$

f = “Haylett” adjustment factor e.g. 0.85

C'_{nj} = actual, inflated cost of item j in month n

The assumptions made to keep this formulation simple continue to include the following:

- that there is no practice of retention, and,
- that the costs of any item j are incurred simply simultaneously with the receipt of the interim payment for that same month.

4.6 COMPLEX COMPOSITE LOADING

Equations 4.1, 4.2 and 4.3 can be combined and is presented as Equation 4.4 below.

Equation 4.2 can be thought of as the same as Equation 4.1 with the addition that the one assumption as regards quantities remaining fixed being addressed. Thus, for these purposes Equation 4.2 can be regarded as an enhanced version of Equation 4.1. To combine these three equations, one therefore only has to combine Equations 4.2 and 4.3.

These can be added together as follows:

$$PV_j = \sum_{n=1}^N \left(\frac{1}{1+r_j} \right)^n \lambda_{nj} (Q_j + Q'_j) (P_j - C_j) + \sum_{n=1}^N \left(\frac{1}{1+r_j} \right)^n [(\lambda_{nj} (Q_j + Q'_j)) [\Upsilon_{nj} f P_j - C'_{nj}]] \quad (4.4)$$

Thus

$$PV_j = \sum_{n=1}^N \left(\frac{1}{1+r_j} \right)^n \lambda_{nj} (Q_j + Q'_j) (P_j - C_j + \Upsilon_{nj} f P_j - C'_{nj}) \quad (4.5)$$

which is

$$PV_j = \sum_{n=1}^N \left(\frac{1}{1+r_j} \right)^n \left[\lambda_{nj} (Q_j + Q'_j) \left((1 + \Upsilon_{nj} f) P_j - C''_{nj} \right) \right] \quad (4.6)$$

where $C''_{nj} = C_j + C'_{nj}$

C''_{nj} = actual, inflated unit cost of item j in month n

This formulation still incorporates the assumptions that applied to equation 4.3 above.

Equation 4.6 can now be adjusted in order to incorporate the effects of retention:

$$PV_j = \sum_{n=1}^N \left(\frac{1}{1+r_j} \right)^n \left[\lambda_{nj} (Q_j + Q'_j) \left((1 + \Upsilon_{nj} f) (1 - R_n) P_j - C''_{nj} \right) + R'_n \right] \quad (4.7)$$

where R_n = proportion retained in month n

R'_n = the amount (if any) released from the retention fund in month n
including any interest earned (if applicable)

Notice that Equation 4.7 takes the form of a linear equation:

$$PV_j = \alpha_j + \beta_j P_j \quad (4.8)$$

where $\alpha_j = \sum_{n=1}^N \left(\frac{1}{1+r_j} \right)^n \left[-C''_{nj} \lambda_{nj} (Q_j + Q'_j) + R'_n \right]$

and $\beta_j = \sum_{n=1}^N \left(\frac{1}{1+r_j} \right)^n \left[\left(\lambda_{nj} (Q_j + Q'_j) \right) \left[(1 + \Upsilon_{nj} f) (1 - R_n) \right] \right]$

with slope $\beta_j = \frac{\Delta PV_j}{\Delta P_j}$

and with ΔPV_j as the change in PV_j

$$\Delta PV_j = \sum_{n=1}^N \left(\frac{1}{1+r_j} \right)^n \left[(\lambda_{nj} (Q_j + Q'_j)) \left[(1 + \Upsilon_{nj} f) (1 - R_n) \Delta P_j \right] \right] \quad (4.9)$$

It is noteworthy that item price loading can do nothing to change α_j , which is the fixed intercept regardless of P_j . (The interest on retained funds may be marginally affected by front-end loading, in particular, but it is suggested that this is of such small consequence that it should be considered irrelevant.) The focus of attention for any item price loading model needs therefore to be on the slope β_j .

The significance of this observation is that this aspect of the overall item price loading model can ignore all considerations of costs (as represented by C_j and C'_j). This furthermore goes to imply that the effect of having the objective of maximising a project's revenue will be the same as having the objective of maximising its profit.

The slope β_j is equal to ΔPV_j if we let $\Delta P_j = 1$, thus

$$\beta_j = \sum_{n=1}^N \left(\frac{1}{1+r_j} \right)^n \lambda_{nj} (Q_j + Q'_j) (1 + \Upsilon_{nj} f) (1 - R_n) \quad (4.10)$$

where β_j is the sensitivity of an item's contribution PV_j to an item's price P_j .

This composite formulation (incorporating the standard forms of unbalanced bidding, as dealt with individually above) provides an important means to identify the relative worth of each of the unbalanced bidding techniques. For instance, it provides a basis by which to determine whether β_j for a particular early item should be relatively high for reason that it is likely to be a valuable contributor to a front-end loading strategy, or instead relatively low for reason that back-end loading should wish to allocate it as low a price as possible. This single, composite formulation presents the means to measure all items, regardless of their unique character, on the same common basis with respect to their potential contribution to a project's overall return.

Let β_j going forward be referred to as the *Profitability Responsiveness Index* (PRI_j). It indicates the sensitivity of an item in terms of the extent to which this item's profit contribution (PV_j) can be improved by way of an increase to its unit price (P_j). If one ignores the issue of risk (dealt with instead in the next chapter) the item j having the highest PRI_j should be the one that has the greatest cause to be allocated the highest price.

Note that it is the relativity of the items' PRI_j that is significant, not the underlying absolute values of the PRI_j s themselves. The PRI_j s facilitate that a contractor can rank a project's items in order of their PRI_j values (giving him their $PRIR_j$ relative ranking). This ranking identifies the relative significance of a project's items in terms of which items will contribute the most profit for any increase in unit price.

To apply this equation (in isolation whilst ignoring for the purpose of this exercise, all other considerations like risk – dealt with in the next chapter), contractor who have the objective of maximising their profit should look to allocate as high a price as possible to the item with the highest $PRIR_j$. Having priced this item as high as possible, they should then look to price the item with the next highest $PRIR_j$ with the highest price possible, and so on, working their way through to the item with the lowest $PRIR_j$ ranking.

4.7 TENDER PRICE CONSTRAINT

The objective of maximising the contractor's profit suggests that, if there were not any constraints, the most profit (in particular, an infinite amount of profit) could be accomplished if a contractor were free to price the item with the highest $PRIR_j$ with an infinitely high price. An infinitely high profit could, in fact, be accomplished by pricing any of the items with an infinitely high price. There are, however, obvious constraints that govern this situation that need to be incorporated into any such mathematical model.

Clearly, it is obvious that one constraint is needed to ensure that the summation of the priced items is the same as the project's overall tender price. This constraint can be expressed by way of the following formulation:

$$\sum_{j=1}^J Q_j P_j = TP \quad (4.11)$$

This same nature of constraint has been incorporated into the models proposed by Stark (1968, 1972, 1974) and Diekmann *et al.* (1982).

4.8 PRICE RATE CONSTRAINTS

It has been commented on by Stark (1968, 1972, 1974), Diekmann *et al.* (1982), Teicholz and Ashley (1978), Tong and Lu (1992), and Christodoulou (2008) that individual item prices should be bound by constraints.

They all expressed these constraints in the following format:

$$L_j \leq P_j \leq U_j \quad (4.12)$$

where L_j and U_j are specified lower and upper bounds, respectively, for the unit price of item j .

Heuristics certainly support the view that some nature of constraint is needed to avoid such extreme situations as having one single item being allocated such a high price that all other items are given the price of *nil*. However, it is questionable whether the best way

to formulate the objective of these constraints is with the use of L_j and U_j as fixed, non-negotiable, constants.

The use of L_j and U_j as fixed constants suggests that a contractor would never consider pricing item j at the price $L_j - 0.01$ or at $U_j + 0.01$ regardless of the merits of doing so. This is patently unrealistic. It is surely more practical to imagine that a contractor may consider using a price of $L_j - a$ or $U_j + a$ where a is a small positive number, should there be sufficient cause for doing so. On this basis, it is impossible to contemplate the justification for the use of any constant values for L_j and U_j no matter how extreme, or how far apart, they are.

These constraints have considerable significance, as will be demonstrated in Chapter 9. In the instances of the models formulated by Stark (1968, 1972, 1974), Diekmann *et al.* (1982), Teicholz and Ashley (1978), and Tong and Lu (1992) these constraints have the effect of almost all items being allocated prices at either their upper (U_j) or lower (L_j) bound limits.

In all of these models, the researchers who formulated them all felt that contractors should have some intuition that can guide them so as to decide the constant values assigned to U_j and L_j . None of them advocated any scientific basis by which a contractor could decide the values of U_j and L_j and yet it is hypothesised that it is these decisions that are the most significant, and yet the most difficult to accomplish, in any of these item price loading strategies.

The proposed model instead advocates that there should be no need for such item price constraints. It is instead suggested that the underlying rationale for such constraints is one of risk. If an item were to be priced exceptionally high (i.e. beyond what any of the above-mentioned models might have capped with an upper limit of U_j) or else exceptionally low (below what these models might have capped by way of L_j) then, it is argued, this would give cause for extraordinary and unacceptable levels of risk. Such risks might include the overall tender being rejected by the quantity surveyor (on the

grounds that the item pricing is unacceptable). Another risk will be from variation orders where an exceptionally high-priced item may be reduced in scale, or else totally eliminated, whilst a low-priced item may be vastly increased in scale. The next chapter will deal with this in more detail.

4.9 SUMMARY

This chapter has provided a comprehensive basis by which to quantify an item's potential contribution to a project's overall profitability. The basis proposed incorporates all three known effects of item price loading: namely front-end loading, individual-rate loading, and back-end loading.

It furthermore has provided a basis by which to measure the sensitivity of a project's overall profitability to adjustments in each of the project's composite item prices. It has proposed that this measure be referred to as each item's *Profitability Responsiveness Index (PRI_i)* and it is this index that determines the relative significance of each item's price (when solely focussed on a project's return and not its risk).

Notice that this model is formulated so as to maximize a project's revenue and not its profit. It was been shown that the maximization of a project's revenue accomplishes the same effect as when the objective is instead to maximise a project's profit. The models of Teicholz and Ashley (1978) and Diekmann *et al.* (1982) are instead structured to take a project's costs into account and hence to maximise the profit. It has been shown that this is more difficult to accomplish and that there is no benefit to be derived from this added complexity.

The research has, furthermore, found that a constraint is required, as with many previous models, so as to ensure that all of the item's prices add up to the project's total tender price. It has, however, also been suggested that individual item price constraints should not be incorporated into the model, even though such constraints have been popularly advocated in all prior item price loading models.

Whilst this chapter has proposed a basis by which the revenue aspect of item pricing can be modelled, the next chapter will cover all aspects of the risks associated with this.

REFERENCES

- Afshar, A. and Amiri, H.** (2008) A fuzzy-based model for unbalanced bidding in construction. First International Conference on Construction in Developing Countries, August 4-5, Karachi, Pakistan. Available at <http://www.neduet.edu.pk/Civil/ICCIDC-I/Complete%20Proceedings.rar#page=68> [accessed 8th April, 2009]
- Abdel-Razek, R.H.** (1987) Computerised analyses of estimating inaccuracy and tender variability: causes; evaluation and consequences. Unpublished PhD thesis, Loughborough University of Technology.
- Cattell, D.W.** (1984) A model for item price loading by building contractors. Unpublished paper, Department of Quantity Surveying, University of the Witwatersrand, Johannesburg.
- Cattell, D.W.** (1987). Item price loading. *PACE'87 Progress in Architecture, Construction and Engineering proceedings*, Johannesburg, July, Vol.II, Session 3, pp. 1-20.
- Cattell, D.W., Bowen, P.A. and Kaka, A.P.** (2004) A model to distribute mark-up amongst quotation component items: an outline, *Proceedings of the 2nd Postgraduate Conference on Construction Industry Development*, Cape Town, Oct., CIDB, pp. 154-165.
- Cattell, D.W., Bowen, P.A. and Kaka, A.P.** (2007) Review of unbalanced bidding models in construction, *American Society of Civil Engineers, Journal of Construction Engineering and Management*, vol. 133, no. 8, Aug., pp. 562-573.
- Christodoulou, S.E.** (2008) A bid-unbalancing method for lowering a contractor's financial risk. *Construction Management and Economics*, vol. 26, pp.1291-1302.
- Diekmann, J.E., Mayer, R.H. Jr. and Stark, R.M.** (1982) Coping with uncertainty in unit price contracting, *Journal of the Construction Division, American Society of Civil Engineers*, vol. 108, no. C03, Sep., pp. 379-389.
- Gates, M.** (1959) Aspects of competitive bidding. *Annual Report*, Connecticut Society of Civil Engineers.

- Gates, M.** (1967) Bidding strategies and probabilities. *Journal of the Construction Division, American Society of Civil Engineers*, vol. 93, no. C01, Proc paper 5159, Mar., pp. 75-107.
- Joint Building Contracts Committee (JBCC)** (2005) JBCC Contract Price Adjustment Provisions (CPAP): CPAP Indices Application Manual. *JBCC Series 2000*, JBCC, May.
- Stark, R.M.** (1968) Unbalanced bidding models – theory. *Journal of the Construction Division, American Society of Civil Engineers*, vol. 94, no. C02, pp. 197-209.
- Stark, R.M.** (1972) Unbalancing of tenders. *Proceedings of the Institute of Civil Engineers*, Technical Note 59, vol. 51, pp. 391-392.
- Stark, R.M.** (1974) Unbalanced highway contract tendering. *Operational Research Quarterly*, vol. 25, no. 3, pp. 373-388.
- Teicholz, P.M. and Ashley, D.B.** (1978) Optimal bid prices for unit price contract. *Journal of the Construction Division, American Society of Civil Engineers*, vol. 104, no. 1, March, pp. 57-67.
- Tong, Y. and Lu, Y.** (1992) Unbalanced bidding on contracts with variation trends in client-provided quantities. *Construction Management and Economics*, vol. 10, pp. 69-80.

5. MODELLING A PROJECT'S RISK

5.1 INTRODUCTION

In Chapter 3 it was shown that the prior research on unbalanced bidding models has been largely focused on the optimization of the expected profits for a contractor rather than too much consideration being given to the risks involved. Researchers have, however, often acknowledged these risks but, despite this, they have made little effort to properly incorporate these risks into their models (see Stark, 1968, 1972, 1974; Diekmann *et al.*, 1982; Teicholz and Ashley, 1978; and Tong and Lu, 1992).

Moreover, besides the work of Diekmann *et al.* (1982), little has been done to structure these models so that they recognize the inherent nature of the trade-off that exists between unbalanced bidding's contributions to these risks and that of the prospective gains. Diekmann *et al.*'s (1982) efforts in this regard, whilst somewhat limited from the point of view that they only recognized one type of risk and then required that a contractor arbitrarily decide on the value of a constant by which to shift the objective from the maximization of profit to the minimization of risk, were nevertheless pioneering.

As previously noted, all models have otherwise given recognition to risk by constraining prices using an imposition of lower and upper bounds to each and every item price (see Stark, 1968, 1972, 1974; Diekmann *et al.*, 1982; Teicholz and Ashley, 1978; Tong and Lu, 1992, and Christodoulou, 2008). It has been observed that these bounds have ended up becoming the single most important aspect of these models, with the effect that all items are priced at either their upper price limit or else their lower price limit, with the exception of only one item. With these models, this one remaining item price serves to ensure that the summation of all the priced items equals the tender price. The effect of these models was therefore reduced to only serving to split all the items into these two groups: those priced high and those priced low.

Early unbalanced bidding models (see Stark, 1968, 1972, 1974; Diekmann *et al.*, 1982; Teicholz and Ashley, 1978; and Tong and Lu, 1992) did not quantify the risks that are generated by way of unbalancing a bid, albeit that all of them have vaguely acknowledged some of the risks and attempted to avoid them. Recent research by Christodoulou (2008) proposed minimizing the *financial entropy* (disorder) of a project as a measure of some aspects of the risk. This research does not, however, provide a comprehensive measure of all of the risks that are involved. Other recent research by Afshar and Amiri (2008) also attempts to address the issue of some of the risks by way of using a Fuzzy Linear Programming approach.

The purpose of this chapter is to establish a basis by which unbalanced bidding models are now able to embrace risk to the same extent, and with the same priority, as that of the expected returns. More particularly so, the purpose is to incorporate this assessment of risk into the model that is proposed by way of this thesis (as formulated in Chapter 8). When a contractor is able to assess both the returns as well as the risks of alternative pricing regimens, they can then begin to apply the principles advocated in chapters 6, 7 and 8 – namely, decide on the pricing schedule that offers them the optimum mix between return and risk.

This chapter is structured that it first gives consideration to the identification of the risks that are involved, then describes how the three types of risks that are identified comprise two different forms; it then describes a Value-at-Risk method of assessing risk and proposes that this provides a solution for combining these two forms, and then describes how these risks can be measured in a manner that is of practical use within this domain.

5.2 THE RISKS OF ITEMS PRICING

There are many risks that are either the direct result of item pricing or that are affected by item pricing. These risks may be classified as: the risk of rejection; the risk of reaction; and the risk of being wrong. Dealing with each type in turn:

5.2.1 The risk of rejection

A priced bills of quantities stands the risk of being rejected, particularly for reason of being unbalanced. If any single item price is either too high or too low it could serve to trigger this rejection. If any one item price is either extremely low or else extremely high, the contractor can be almost 100% certain that the client will reject their bid. Wang *et al.* (2006), Touran and Ghavamifar (2008), and Arditi and Choktibhongs (2008) show that clients are not known to use any sophisticated methods of quantification by which to assess bids with a view to rejecting those that might be unacceptably unbalanced.

Without any such techniques by which to assess item prices collectively, it is hypothesized that clients instead judge bids on the basis that one or more of the component item prices stand out as an outlier. It is assumed for the purposes of this research that any one extreme item price may be sufficient to potentially spoil the whole of a contractor's bid. It is being assumed that a bid stands to be at risk of being rejected if only one of the item prices is as low as *nil* or else so high as to be several orders of magnitude larger than what might appear to be reasonable. Only one such price may be sufficient to trigger the perception that the pricing of the overall project is unacceptable and therefore to be rejected. This assumption should be the subject of further research: it is alternatively possible that a more sophisticated method may be appropriate, for instance, so as to 'bundle' some items together in this regard, rather than treat them independently.



Figure 5.2 Risk to price relationship for a typical item

This function of risk to price is shown in Figure 5.1, resembling an inverted, skewed normal distribution, split on the X-axis by a central range in which the risk is so small that it is effectively *nil*. Notable characteristics of this function are that the risk of rejection at any extreme price is almost 100% and the risk around some central “reasonable price” is infinitesimally small. The difference between this function for each of the items, is capable of being described by way of a combination of the following four attributes:

- a. the point along the x-axis (being the mean) where the price is least likely to be objectionable,
- b. the width of the central range at which the risk is treated as effectively nil,
- c. the variance (or ‘spread’ or ‘width’) of the curve outside of the central range, and
- d. the extent to which the curve is asymmetric (i.e. skewed) so as to account for increased price sensitivity as the price approaches its least value of *nil*.

Heuristics suggest that different items have risk curves with differences in these characteristics such as the above-mentioned ‘spread’. For example, items of excavation are likely to have a ‘wider’ range of tolerance than items that have a greater certainty

such as painting items (see Beeston, 1975). An item that has characteristics that are inherently less certain, such as any of those related to earthworks, is likely to ensure that the client is more tolerant of a wider range of prices before they are likely to reject the overall bid. This is shown in Figure 5.2.



Figure 5.2 Risk to price relationship for an item of relative uncertainty

On the other hand, there are some items where the contractor can be almost certain that the client will not be accepting of any price other than one possibility, and where any other price will likely arouse suspicions and rejection. Examples of this include Prime Cost and Provisional Sum items, which cannot be priced any differently than that which is directed in the bills of quantities. If a provisional sum item requires that (say) R10,000.00 be set aside for an, as yet, ill-defined item of work, and if this is to include a stated fixed percentage allocation to profit and attendance, the contractor does not in effect have any freedom by which to price this item in any other way. This type of situation is graphically depicted in Figure 5.3.

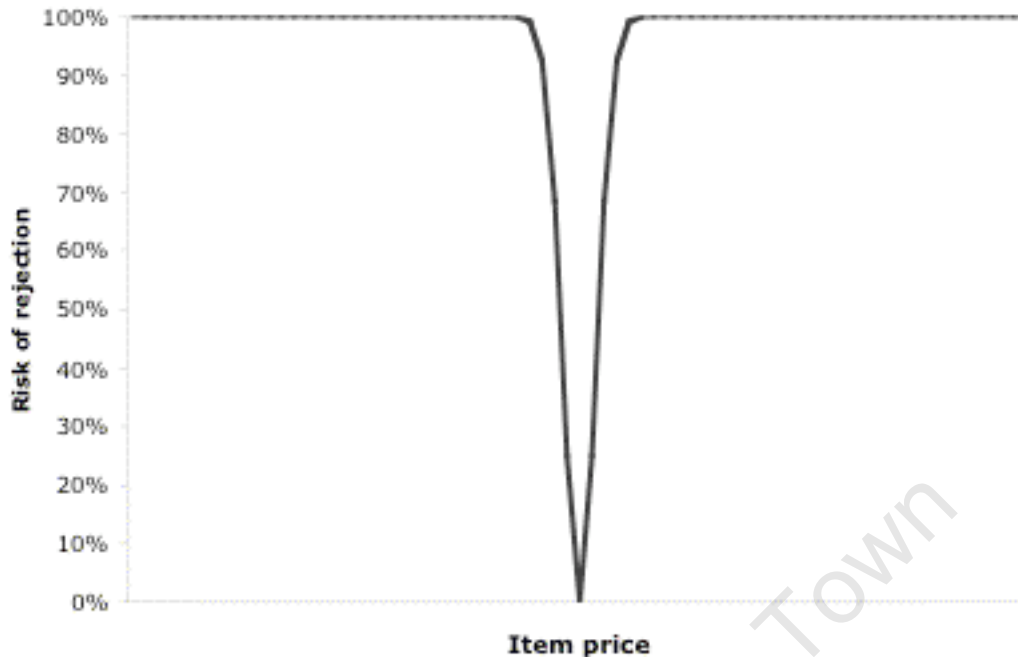


Figure 5.3 Risk to price relationship for an item of relative certainty

Various efforts have been made (see Wang *et al.*, 2006) to develop models to assist clients with identifying unbalanced bids. Typically, these are dependent on comparing a contractor's prices against a benchmark and then concluding that if these differ by too much, then the bid is likely unbalanced. The problem with this approach rests largely on their dependence on these benchmarks that are themselves potentially (and likely) unbalanced, thus making any such comparison less than objective. The reason for these benchmarks being likely unbalanced is their dependence, in turn, on previous bids which themselves are, quite possibly, unbalanced. These models might therefore serve to assess only whether a bid is *extraordinarily* unbalanced and less so serve to determine conclusively that any bid has some (more normal) degree of unbalancing.

In practice, the use of such a model has yet to become popular. Instead, it is almost certain that clients (and their professional agents) will be 'looking out', using less formal methods, for any item prices that appear either extraordinarily high or low. For this reason, two methods have become popular as techniques by which to constrain unbalanced bidding models, namely:

1. upper and lower price limits for each item

$$L_j \leq P_j \leq U_j \quad (5.1)$$

where P_j = item price for item j

L_j & U_j = lower and upper bounds, respectively, for the price of item j

It was noted in section 4.8 that absolute limits of this form are heuristically inappropriate. Instead, it was shown that prices should be bounded by limits that are more fuzzy than fixed.

2. inter-item pricing limits

$$P_i - P_j \geq 0 \quad (5.2)$$

where i & j = different item numbers

In the same way that it has been recognized that item prices are having to be bound without referring to each other, Stark (1968) identified that the prices for some items should be made to be the same or higher than other items. For example, he argued that the price for the excavation of hard-rock should logically be more than the price for the excavation of soft-rock, else it shall be obvious that the bid has been unbalanced.

Heuristics suggest that the risks of rejection for all of a project's items are not strictly additive. It is hypothesized that it is only necessary for a single item to be rejected in order for this to trigger a rejection of a contractor's overall project bid. The risk of rejection for the project is therefore the same as that of the most objectionable item price - in other words, the one with the highest risk of rejection. It is suggested here that, even if the prices for all of a project's other items were well within the normal bounds of expectation, that the overall bid nevertheless stands the risk of being rejected if only one single price, an outlier, is sufficiently excessive (either low or high). This assumption

requires further research and, in particular, as regards whether it is possible that clients may be less sensitive to low-value items than high-value items. It is also possible that it may be found that clients may be more sensitive to items that are frequently common in projects than with items that are more unusual or unique to a project. For instance, it is hypothesised that clients may be found to be especially on the 'look-out' for the pricing of brickwork and excavations and other items that many projects have in common, and that are easily comparable between projects.

It is necessary to acknowledge here that some development environments may be more 'ruthless' than others. For instance, in the public sector, clients may be more obliged to be transparent as regards being fair to all those who bid, by imposing a high standard by which they will reject a priced bills of quantities. In such a situation, it is possible that clients may not wish to negotiate any compromise on any item prices and they may be under more political pressure of scrutiny to outright reject any loading of the bills, if this is detected by them as being obvious. However, in other situations, contractors could likely anticipate that clients are less likely to reject them and more likely to respond to 'loaded' prices by way of insisting that these be adjusted. If the latter environment applies, the contractor may wish to record that the value of the 'risk of rejection' is effectively as little as *nil*, even though they anticipate that some pricing scenarios may nevertheless be rejected. A *nil* value of rejection will be an acknowledgement that the worst-case outcome that they anticipate will be one of being forced into a negotiation in which they are asked to moderate the loading of some prices.

5.2.1 The risk of reaction

If an item's price is very high, it might inspire an architect to redesign the project so as to avoid this item. Similarly, if an item's price is thought to be attractively low, it might give rise to a variation order in which the project uses more of this item. For example, if the price of face-brick walls is low relative to alternative finishes then the architect might take this as an opportunity to switch finishes to take advantage of this pricing; perhaps by giving consideration to both the initial cost as well as likely maintenance expenditure.

This is a risk that the contractor should factor into their item pricing strategy.

5.2.2 The risk of being wrong

Unbalanced bidding models incorporate many variables that require that the contractor has to estimate, and these estimates are inevitably, to some degree, wrong. Any such errors obviously lead to the generation of item prices that might be different had the contractor not been wrong. In other words, if, at the time of pricing the bills of quantities, a contractor could enjoy the accuracy of hindsight then their item prices are likely to be different than those that they produced on the basis of their (imperfect) estimates.

In particular, the item pricing model provided in Chapter 4 requires that the contractor provide estimates for the following variables: a discounting rate; the scheduled timing of items; any expected variation in quantity (such as if the contractor thinks that the site contains more rock than is contained in the bills of quantities); and escalation rates (for each workgroup).

It is interesting to observe that, although variables such as the discounting rate are common to all items, the risk associated with the variability in estimating this rate is different for each item. Items scheduled at the beginning of the project are hardly discounted whilst items scheduled at the end of the project are heavily discounted. Furthermore, the later an item is scheduled to occur, the greater the chance that the estimate of the rate will be wrong.

5.3 TWO FORMS OF RISK

The above-mentioned risks take one of two forms: risks that arise as a direct response to the price in question; and those that will occur regardless. In the former instance, the price is the cause that gives effect to the risk, whereas with the latter form of risk, the price does not trigger the risk and the risk conditions remain systemic regardless of the price. In this latter instance, even though the price may not give cause for the risk, the extent of the risk is nevertheless a function of the price: higher prices give contractors a higher exposure to such risks.

These risks are hereby defined as the *direct* and *indirect* risks, respectively. The risks of *rejection* and *reaction* are therefore direct risks: excessive prices (high or low) give rise to high levels of such risk, whereas ‘normal’ prices may give rise to infinitesimally small risk. The *risk of being wrong* is, however, by this definition, an indirect risk: the conditions that make the task of estimating and forecasting a challenge remain systemic regardless of the contractor’s prices. Nevertheless, if these conditions are especially worse (i.e. more risky) for one item than other items, contractors will expose themselves to more risk by giving a high price to such an item. Loading the price of a risky item and then, as a result of a need to fund this, correspondingly unloading the prices of less risky items, increases the contractor’s overall risk. For instance, if conditions are such that it is exceedingly difficult to estimate the amount of rock excavation at a site, then the contractor’s risk will be amplified if they allocate high prices to the items associated with this rock excavation.

With *direct* risks it is suggested that the price at which this risk is least is not as much having to be influenced by any estimate of the cost of this item as it is having to be positioned relative to item prices that the industry in general has grown accustomed to with respect to this nature of item. For instance, if clients have become accustomed to high costs of excavation (perhaps by reason of an industry-wide systemic prevalence of front-end loading) then the bell-curve of such items should be centred around that expectation.

5.4 VALUE-AT-RISK

What is desired is a way by which to measure all of the risk (namely, all of the above-mentioned risks, combined) generated at each level of price for each item of a project. With this, one will then know both the rewards that are to be expected for all of these price levels and also the risks. If these are structured so that they ‘interlock’ (i.e. that they are additive) with those from other items, and if one recognises that all of the item prices must add up to the overall bid price, then one has a basis by which to measure the overall expected reward and risk from all possible item price combinations.

To reach this goal, one single measure of all of the risks combined is required. However, the above-mentioned analysis has already identified that the overall risk from any item comprises two very different forms of risk: the one normally suited to measurement by way of some assessment of its variance, and the other not. To overcome the challenge of combining these, it is proposed that a very useful, relatively new measure of risk be used that appears to be conveniently well suited to this purpose. This measure is known as 'Value-at-Risk' and is abbreviated as 'VaR' (Manganele and Engle, 2001).

The VaR method is said (Kolman *et al.*, 1998) to have been developed at the J.P. Morgan Bank (J.P. Morgan, 1994) in the late 1980s / early 1990s for the purposes of assessing the bank's exposure to risk on its equity positions. A report by the Global Derivates Study Group (1993) is said to have first published the use of the phrase "Value-at-Risk".

According to Manganele and Engle (2001: 4) the VaR measure has gone on to "*become the standard measure that financial analysts use to quantify market risk. VaR is defined as the maximum potential change in value of a portfolio of financial instruments with a given probability over a certain horizon.*" Typically this time horizon is only one day and the probability used is often 1% (Benninga and Wiener, 1998). A common form of expressing the VaR of an investment portfolio is therefore along the lines that it has been assessed that there is only a 1% risk that the value of a portfolio could drop by more than (say) R1m within the next day (Jorion, 2006). Expressed another way, one expects that the value of this portfolio will erode by more than R1m within 24 hours, only as often as 1 day in 100. A lot of the appeal of this method would appear to lie in the simplicity by which risk can be expressed and comprehended (Schachter, 1997).

This method of expressing risk lends itself well for the purposes of unbalanced bidding, rather than the more traditional methods of expressing risk in terms of variance (for the traditional approach, see Edwards, 2001). As an example, it facilitates that an unbalanced bidding model might predict that there is a 50% risk being generated by way of an item being assigned a price of R1.00 that, as a direct result of this, the contractor might lose (say) R10,000. Furthermore, this model might predict that if this item is priced at R1.10 then there is a 50% risk that the contractor might lose (say) R8,000. If the contractor can be this well-informed and have this degree of insight into their item

pricing, they would be well-equipped to price their items to not only maximize their expected reward, but also to minimize their risk.

Notice that this methodology facilitates that contractors no longer need to implement the item price constraints that are of the form of Equation 5.1. As discussed (in Section 4.8) these constraints have been identified as the single most important influence on almost all previous unbalanced bidding models, and yet, as has been explained, this is heuristically flawed. Instead, this new methodology of addressing risk overcomes these problems and it gives effect to a more indefinite nature of *fuzzy* bounds rather than the definite fixed bounds which are problematic.

5.5 VALUING THESE RISKS

5.5.1 The risk of rejection

To implement the ideas expressed in Section 5.2.1 above, the contractor needs to estimate the extent of their loss that they will suffer in the event that any item price of theirs leads to a rejection of their overall bid. In addition, they need to estimate the degree of variance by which the aforementioned normal distribution describes the likely response from the client – as shown in Figures 5.1 to 5.3. The equation for a normal distribution takes the following form:

$$\phi(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (5.3)$$

The standard form of this, where $\sigma = 1$ and $\mu = 0$ is therefore as follows:

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \quad (5.4)$$

with the following cumulative distribution function

$$\Phi(x) = \int_{-\infty}^x \phi(t) dt = \frac{1}{2} \left[1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right) \right] \quad (5.5)$$

By comparison, the probability density function of a skew-normal distribution is then as follows:

$$f(x) = 2\phi(x)\Phi(\alpha x) \quad (5.6)$$

where α is a measure affecting the shape of the skew, and in this instance is a positive, real number so as to skew the curve to the right.

To produce the risk function (shown in Figures 5.1 – 5.3) this equation needs to be inverted and standardised so that the range of the risk is scaled to range from 0% to 100% (or infinitesimally close to this, to be theoretically correct). This can be done using the following equation:

$$risk = (y - y_{\min}) / (y_{\max} - y_{\min}) \quad (5.7)$$

The variance (as described in Equation 5.6 by way of the standard deviation σ) is a function of two influences: one of which is item-specific, and the other which is general to the overall project. The item-specific variance relates to the nature of the particular item as described by way of the different scenarios underlying the differences depicted in Figures 5.1 – 5.3 above. For instance, it is hypothesized that prices across the industry for earthwork items have far greater inherent variability than the prices of, say, items of specified ironmongery, painting of general surfaces, prime cost items or provisional sums. Ideally, and drawing on the exploratory work of Beeston (1975), research needs to be done to confirm this hypothesis and variances need to be identified for different categories of items, as reflected in the variability of item pricing between contractors, across a wide range of projects.

The second influence on this variance is dependent on the relative competitive strength of the winning contractor. If the scenario is one where the tender procedure is that the client will choose one of the bidders (typically the lowest one) and ask them to then submit the project's component item prices, both the client and the contractor will have knowledge of the numerous contractors' tender bids at the time at which they are having to decide their item prices. If the contractor's (lowest) bid is only marginally less than the bids of

other contractors, then this places the client in a stronger negotiating position by which to reject any proposed item pricing than if the lowest bid is substantially less than any other. In an extreme situation, if the 'winning' contractor were to realize that their bid had been a mistake (and hence they have submitted far too low a price, threatening perhaps to cause them a loss if the project goes ahead on this basis), it is quite easy to imagine that they will be highly motivated to then submit item pricing that is considerably unbalanced, forcing the client's hand to reject their bid. Even if the client were not to reject this pricing, the exceptional loading of the bid will stand to generate some additional profit that will help make up for some of the loss. In any event, it is not difficult to imagine that the knowledge of all the bids is likely to influence the extent of both the contractor's motivation and the client's 'willingness-to-accept' any unbalancing of the bid.

It is suggested that contractors conduct research on the variability of item prices, specific to different trades across their industry. Implicit in this variability is the knowledge that clients have been found to have deemed these prices as acceptable. This should determine the standard deviation that they will apply to the analysis of the 'Risk of Rejection' of all such item prices for all projects of theirs. Furthermore, they might wish to apply some intuition to either 'dial this up' or 'dial it down', to some overall degree for any project in question, based on their perception of their relative competitive-strength for reason of the spread between their (lowest) bid and that of the second lowest.

The next important aspect of this risk assessment process is the *quantum* of the loss that the contractor will suffer in the event that a client does reject their prices. Heuristics suggest that this loss does not amount to the profit budgeted for the project. If any project does not go ahead, another project will take its place. If a contractor's pricing is rejected, the likely consequence is that they will have to submit several more bids before winning another one. The cost of being rejected is proposed to therefore be representative of the following aspects: the cost of estimating and bidding several more projects, including all those that will be lost before the contractor is again in the position where their bid is the lowest; and accounting for any extraordinary profit / loss that is expected from this project. The contractor might also choose to account for the

attractiveness of the current project as regards its use of the contractor's resources and how well this fits in with the contractor's schedule of other work.

It is hypothesized that the latter issue is again a function of the spread "left on the table" between the lowest bid and the second lowest. If the contractor has regrets that they should have submitted a higher bid, they might perceive the amount of the prospective loss to be minimal. Indeed, they may even perceive that it will be to their advantage to lose this project, i.e. that the prospective loss from doing so is a negative one. With hindsight of all the bids, the contractor could perceive it to be more valuable to them not to have "won" at this level of bid.

The value of 100% of the risk of rejection for any item is the same as for any other. They should all be assigned the same (Value-at-Risk) number in this regard, with the differences between the risk on items being reflected solely by way of different mean prices applying to different items (with this determining the position on the x-axis of the chart for each item, i.e. the same 'expected' price) and the 'width' or variance for each item, as explained above.

On this basis, if the contractor determines that the opportunity cost for them of losing this project is, say, R100,000, and if they have determined that the mean price and standard deviation for a specific item is R10.00 and R1.00, respectively, then the function for determining the risk of rejection for this specific item can be plotted as shown in Figure 5.4 below.

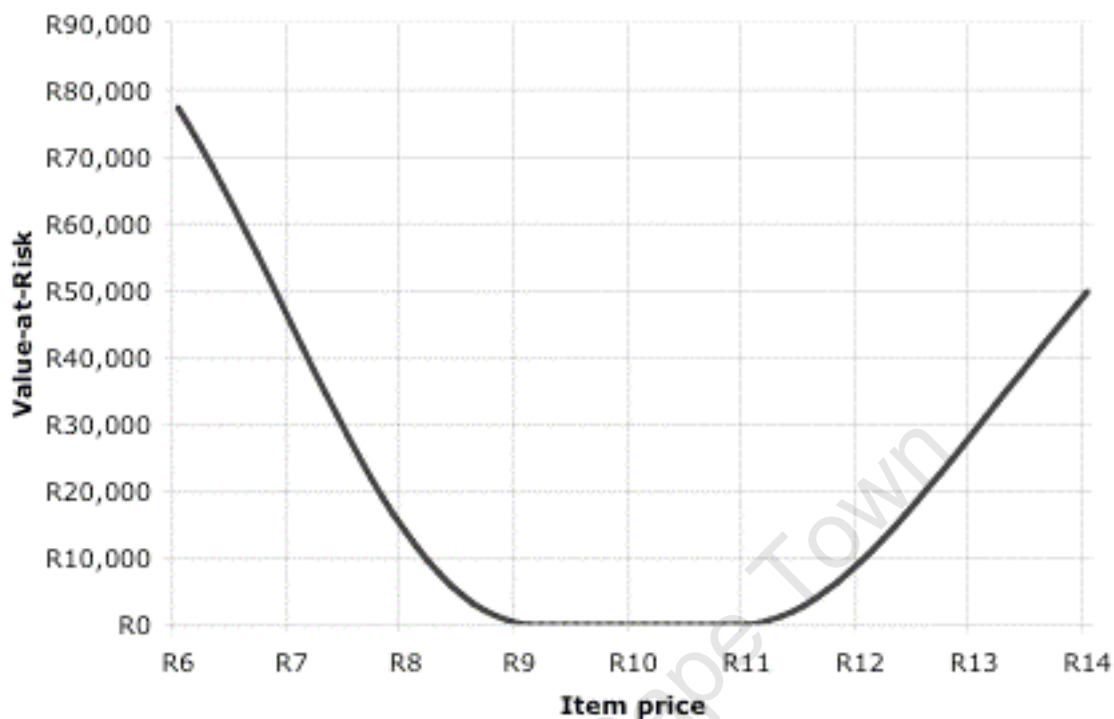


Figure 5.4 Value-at-Risk for reason of the risk of rejection

5.5.2 The risk of reaction

The contractor has a similar need to quantify the risk of inspiring a variation order. For many items this risk may be *nil* seeing as it may not be possible for the client to redesign the project to avoid this item, else to increase its quantity. For other items, it may be considerably more.

By comparison to the (above-described) 'risk of rejection', this 'risk of reaction' does not plateau at any maximum monetary value. The higher the price, the higher the expected loss if the client were to react by way of a variation order.

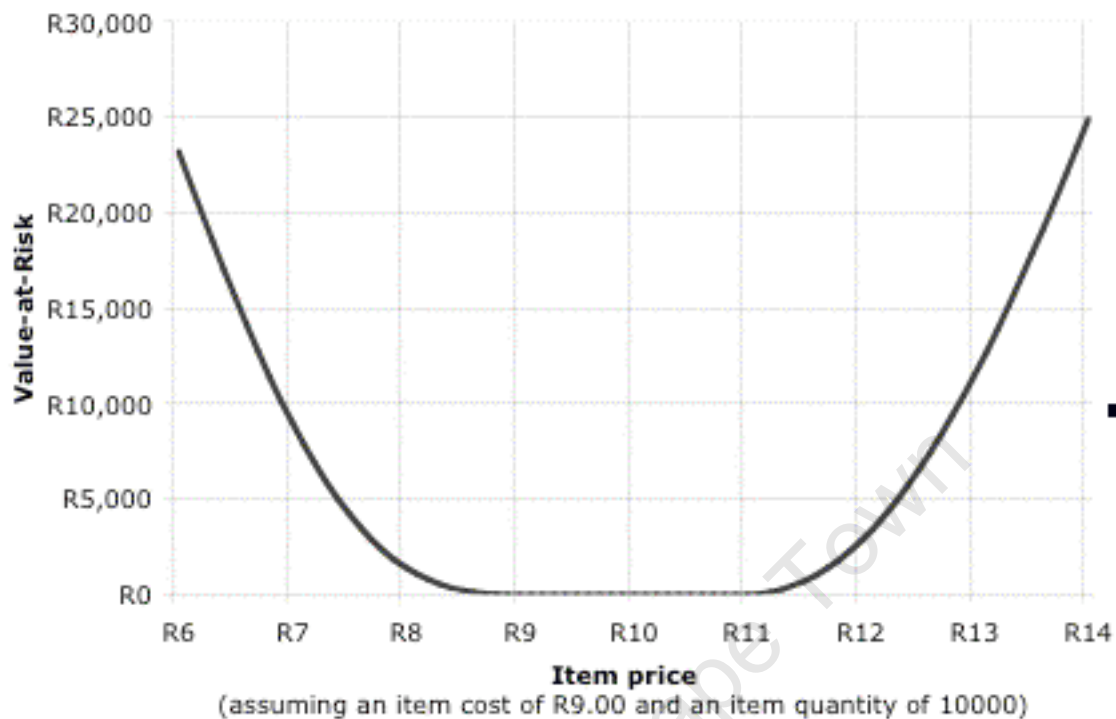


Figure 5.5 Value-at-Risk for reason of the risk of reaction

Figure 5.5 shows how, if the item is priced at less than its cost, then there is a risk associated with the expectation that the quantity of this item may be increased by way of a variation order. Beyond the mean expected price, the risk becomes one of the item being replaced with another. On the right-hand side of this curve, as this risk ascends the inverted, skewed and split normal distribution curve, the *quantum* of the monetary risk is the loss of profit that will be suffered should this item be replaced. Further research is required to add the sophistication of treating items as being 'related' to each other, rather than treating them as all being independent. For instance, if a high price of face-brick walls were to threaten a switch to the alternative finish of plaster and paint, this risk would be more if these alternative finishes were, at the same time, priced low, rather than if they were also priced high.

5.5.3 The risk of being wrong

The monetary *quantum* of the risk of being wrong is a function of the uncertainty in the underlying assumptions and estimates that comprise the computation of the expected return that is to be enjoyed from the activity of item pricing.

Equation 4.10 provides the measure of expected reward from each item j for each unit of currency applied to its price. As more fully explained in Chapter 4, β_j describes the ‘responsiveness’ or ‘sensitivity’ of the contractor’s expected reward to any assigned price. Items with high β_j values are, in other words, more receptive to high prices than items with lower β_j values: if a contractor is only motivated by their prospective reward and disregarding of their risks, they will benefit more by shifting their pricing into the items with higher β_j values and away from the items that have lower-ranking β_j values. This objective is, however, made more complicated when considering that contractors are also, at the same time, inevitably wishing to minimize their risks.

This assessment of β_j is itself not something that can be done with certainty and there is a risk associated with the variability in this estimate. The contractor needs to consider the underlying uncertainty in each of the following estimates that have been utilized: a discounting rate; the scheduled timing of items; any expected variation in quantity; and likely escalation rates (for each workgroup).

By using Monte Carlo simulation (see Chapter 8 for a detailed explanation), the contractor can determine the Value-at-Risk by way of these factors combined. Figure 5.6 has been produced as an example, using a spreadsheet analysis that combines the uncertainties assuming appropriate correlation between these variances if and when these exist. For a contractor to generate any such analysis, it is necessary that they have to make a variety of assumptions as regards the uncertainty / variance in the above-mentioned variables. These will all vary from project to project depending on the prevailing circumstances.

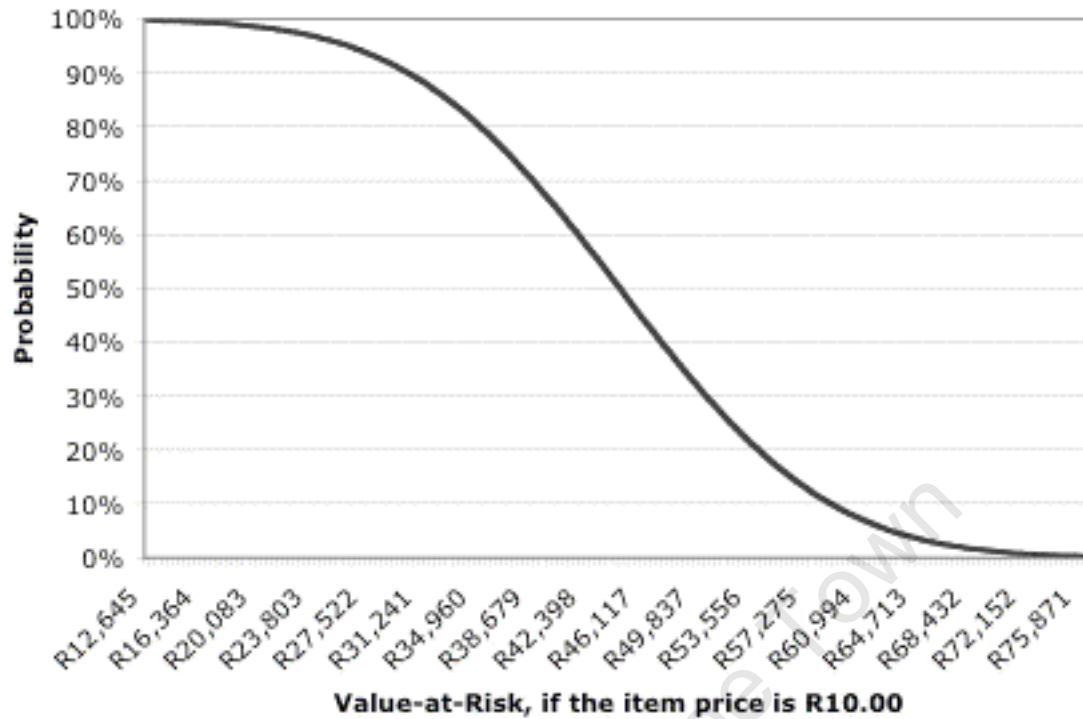


Figure 5.6 Value-at-Risk for reason of estimating uncertainty

Figure 5.6 has been compiled using a spreadsheet that incorporates an inversion of a cumulative distribution function, derived from a normal distribution.

This figure shows the potential loss by ‘being wrong’ as regards estimates of all of the variables being different from their expected (mean) values. For instance, it is shown in Figure 5.6 that there is a 30% probability that, if this item is priced at R10.00, the (present-day) Value-at-Risk, for reason of uncertainty in the above variables, is around R50,000. In other words, it has been calculated that there is a 30% chance that the return from having assigned a R10.00 price to this item will be R50,000 less than that which is expected.

This risk tends to R0 as the probability tends to 100%.

Higher prices for any item result in more risk, the extent to which depends on the inherent uncertainty associated with this item. An example of this is shown in Figure 5.7.

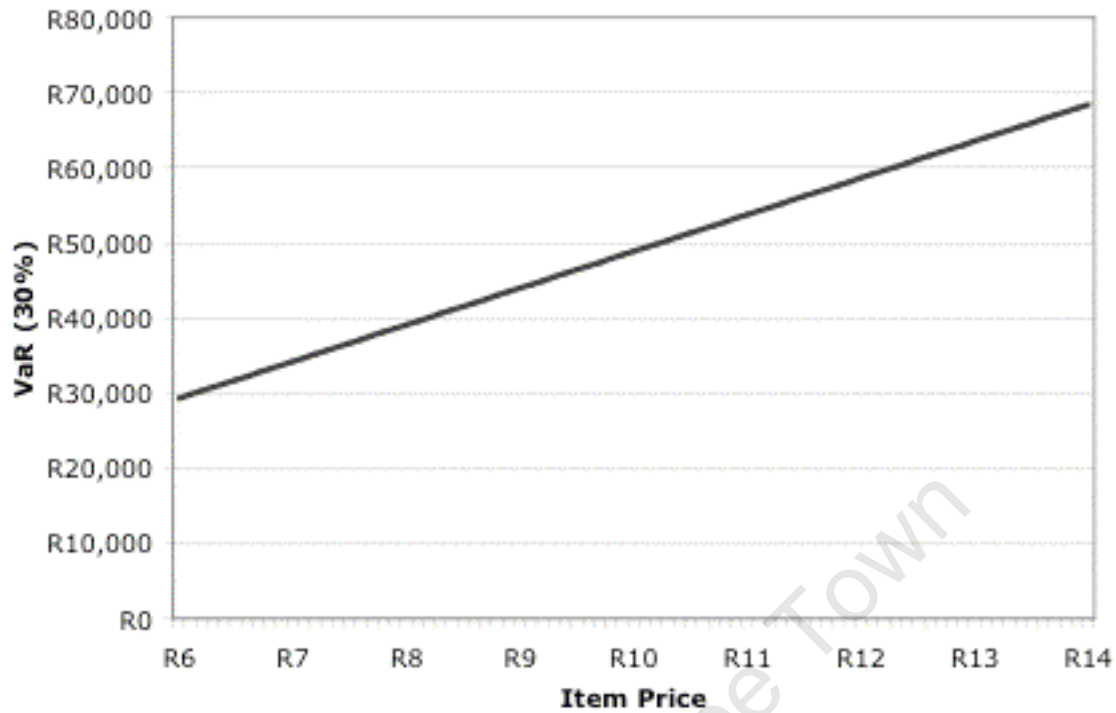


Figure 5.7 VaR as a function of the price assigned to an item

When this risk is expressed in this form it gives a basis by which it can be combined with the risks contributed by way of ‘rejection’ and ‘reaction’.

5.5.4 Illustration of the combined risks

By deciding on a rate of risk for the purposes of this analysis, the contractor can then aggregate the three risks that will be generated by each item at each price point. An example is shown in Figure 5.8.

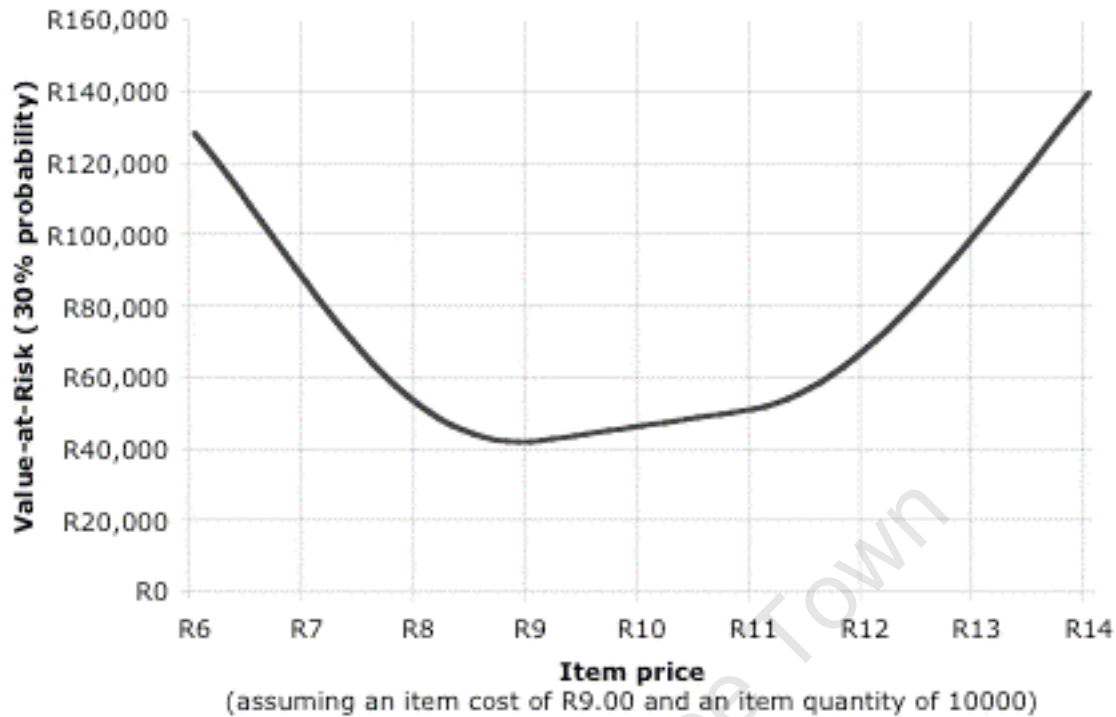


Figure 5.8 VaR, assuming a 30% probability, from all risks combined

5.6 DISCUSSION

Notice that the proposed model is the first to succeed in constraining prices by way of a ‘fuzzy’ nature of boundary recognizing that extreme prices might not only generate high expected returns but also generate considerable risk. This model succeeds in quantifying the need for price constraints in a manner that provides contractors with a tool by which they can weigh up the allure of high returns relative to the risks that are involved. This is illustrated in the results shown later in Chapter 9.

5.7 SUMMARY

This chapter has determined a new framework to recognize the risks that are generated by way of unbalanced bidding. It has also provided a basis by which these can be assessed, combined and incorporated into an unbalanced bidding model. It has found that these risks can be classified as the risks of *rejection*, *reaction*, and of *being wrong*. It has found that these risks are of two different forms but it has proposed that the VaR measure of risk (rather than the more usual use of some measure of variability) can be used as the basis by which to evaluate these risks on the same basis, in such a way that they can be

treated as additive to each other. This summation of the risks, together with the summation of the expected benefits (as described in Chapter 4) provides the basis by which all pricing scenarios can then be evaluated by way of both of these scoring sheets.

In the next chapter, Modern Portfolio Theory will be discussed as one of the ways used in mainstream microeconomics, to address occasions where someone is assessing the relative merits of alternatives that involve different likely returns and different degrees of risk. This shows that a high-return / high-risk scenario is not necessarily preferable to a low-return / low-risk scenario, or vice versa, but that all scenarios that don't offer the highest level of return for the degree of risk involved are not to be regarded as *efficient* relative to the other alternatives and hence not to be considered attractive and worth further consideration by any contractor who is rational.

REFERENCES

- Afshar, A. and Amiri, H.** (2008) A fuzzy-based model for unbalanced bidding in construction. First International Conference on Construction in Developing Countries, August 4-5, Karachi, Pakistan. Available at <http://www.neduet.edu.pk/Civil/ICCIDC-I/Complete%20Proceedings.rar#page=68> [accessed 8th April, 2009]
- Arditi, D. and Chotibhongs, R.** (2008) Detection and prevention of unbalanced bids. Unpublished paper. Illinois Institute Of Technology, Department of Civil and Architectural Engineering, Chicago, IL, USA.
- Beeston, D.T.** (1975) One statistician's view of estimating. *The Building Economist*, December, pp.139-145.
- Benninga, S. and Wiener, Z.** (1998) Value-at-Risk (VaR). *Mathematica in Education and Research*, vol.7, no.4, pp. 39-45.
- Cattell, D.W., Bowen, P.A. and Kaka, A.P.** (2007) Review of unbalanced bidding models in construction. *American Society of Civil Engineers, Journal of Construction Engineering and Management*, vol. 133, no. 8, pp. 562-573.
- Cattell, D.W., Bowen, P.A. and Kaka, A.P.** (2008) A simplified unbalanced bidding model. *Construction Management and Economics*, vol. 26, pp. 1283-1290.
- Christodoulou, S.E.** (2008) A bid-unbalancing method for lowering a contractor's financial risk. *Construction Management and Economics*, vol. 26, pp. 1291-1302.
- Diekmann, J.E., Mayer, R.H. Jr. and Stark, R.M.** (1982) Coping with uncertainty in unit price contracting. *American Society of Civil Engineers, Journal of the Construction Division*, vol. 108, no. C03, pp. 379-389.
- Edwards, P.J.** (2001) A study of risk perceptions and communication in risk management for construction projects. Unpublished PhD thesis, Department of Construction Economics and Management, University of Cape Town.
- Global Derivates Study Group** (1993) *Derivatives: practices and principles*. Washington, Group of Thirty. Available at http://www.group30.org/pubs/pub_0901.htm [accessed 8th April, 2009]
- J.P. Morgan** (1994) *RiskMetrics technical document*. 2nd ed., New York, Morgan Guarantee Trust Company.

- Jorion, P.** (2006) *Value at risk: the new benchmark for managing financial risk*. 3rd ed., New York, McGraw-Hill.
- Kenley, R.** (2003) *Financing Construction: Cash flows and cash farming*. London, Spon Press. Kolman, J., Onak, M., Jorion, P., Taleb, N., Derman, E., Putnam, B., Sandor, R., Jonas, S., Dembo, R., Holt, G., Tanenbaum, R., Margrabe, W., Mudge, D., Lam J. and Rozsypal, J. (1998) Roundtable: the limits of VaR. *Derivatives Strategy*, April.
- Manganele, S. and Engle, R.F.** (2001) Value at risk models in finance. European Central Bank: *Working Paper Series*, no. 75, August.
- Markowitz, H.M.** (1952) Portfolio selection. *Journal of Finance*, vol.7, no. 1, pp. 77–91.
- Markowitz, H.M.** (1990) Foundations of portfolio theory. *Nobel lectures in economic sciences 1981-1990*. Ed. Karl-Göran Mäler, World Scientific, pp. 279-288.
- Schachter, B.** (1997) An irreverent guide to Value at Risk. *Financial Engineering News*, vol. 1, no. 1.
- Stark, R.M.** (1968) Unbalanced bidding models – theory. *American Society of Civil Engineers, Journal of the Construction Division*, vol. 94, no. C02, pp. 197-209.
- Stark, R.M.** (1972) Unbalancing of tenders. *Proceedings of the Institute of Civil Engineers*, Technical Note 59, vol. 51, pp. 391-392.
- Stark, R.M.** (1974) Unbalanced highway contract tendering. *Operational Research Quarterly*, vol. 25, no. 3, pp. 373-388.
- Teicholz, P.M. and Ashley, D.B.** (1978) Optimal bid prices for unit price contract. *American Society of Civil Engineers, Journal of the Construction Division*, vol. 104, no. 1, pp. 57-67.
- Tong, Y. and Lu, Y.** (1992) Unbalanced bidding on contracts with variation trends in client-provided quantities. *Construction Management and Economics*, vol. 10, pp. 69-80.
- Touran, A. and Ghavamifar, K.** (2008) A method for detecting unbalanced bids in unit-price contracts. Unpublished paper. Department of Civil and Environmental Engineering, Northeastern University. Boston, MA, USA.

Tversky, A. and Kahneman, D. (1992) Advances in prospect theory: cumulative representation of uncertainty. *Journal of Risk and Uncertainty*, vol. 5. pp. 297–323.

Wang, W.C., Wang, H.H., Lai, Y.T. and Li, J.C.C. (2006) Unit price based model for evaluating competitive bids. *International Journal of Project Management*, vol. 24, no. 2, pp. 156-166.

University of Cape Town

6. MODERN PORTFOLIO THEORY

6.1 INTRODUCTION

Chapter 4 has described a method by which a contractor can model the reward that they will enjoy from any item price scenario. The approach incorporates the benefits that can be derived from ‘front-end loading’, ‘back-end loading’ and ‘individual rate exploitation’ so that different item pricing combinations can be compared with a view to finding the one combination amongst these that promises to produce the greatest profitability for the contractor. On this basis alone, it would be logical that a contractor should seek to price a project’s items so as to accomplish the greatest profits in the process.

This approach has been taken by all unbalanced bidding models, until Diekmann *et al.* (1982) identified that contractors might also wish to price projects in such a way so as to also reduce their risk.

All research in this field has, to date, been vague as regards what constitutes these “risks” of item pricing and all the models that have been proposed have largely solely depended on imposing minimum and maximum pricing limits on each of the items as the sole technique by which to keep these limits within an (unquantified and undefined) realm of “acceptable risk”. The current research, however, has now assessed these risks (in Chapter 5) and it has laid down a framework by which these risks can now be identified as the risks of “rejection”, “reaction” and of “being wrong”. Chapter 5 has shown a new technique by which these risks can now be measured with respect to any particular item pricing scenario.

Thus, we now have a situation where we can assess any item pricing scenario for both the profitability that this is most-likely to generate, as well as the risks that will be associated with this. Given that a contractor can have this information on-hand, this facilitates that they can seek a scenario that offers them their best-suited trade-off between return and

risk. It is logical that some scenarios may offer high returns and high risks, and others low returns and low risks. These could all be considered contenders for the best-suited trade-off, whereas low-return, high-risk scenarios cannot be considered to be attractive as long as there are alternatives.

This nature of choice is similar to the one that faces many investors and so it has been decided to investigate the science of such situations, when they arise in the mainstream environment of financial economics. Harry Markowitz has pioneered a study of Modern Portfolio Theory which has much to contribute in this regard. This Chapter 6 outlines this theory and the next chapter describes another mainstream micro-economic theory that also has much to contribute to this research - namely that of Cumulative Prospect Theory. These theories, combined, are then applied in Chapter 8 where software is described that has been developed to test the resultant composite Component Unit Pricing Theory (CUP theory) on a hypothetical project. These results are discussed in Chapter 9.

The current chapter serves only to provide a brief overview of MPT and to explain how it has served to provide a conceptual framework as regards scenarios that entail the following characteristics:

- they all entail compromises between prospective returns and the risks thereof,
- these alternatives comprise combinations of constituent components that can be assembled in varying proportions so as to influence the overall returns and risks, and,
- the situation lends itself to the concept and identification of *efficiency* as regards the ability to discard the vast majority of the possible combinations in favour of a small (*efficient*) subset that are all worthy of further consideration.

MPT is a substantial field of research, that includes the mathematical modelling and solving of these problems which was also advocated by Markowitz. He suggested using the Simplex Algorithm, that had been discovered shortly before MPT, in 1947, by Dantzig (1949). The Simplex Algorithm, which has now become a popular basis of solving linear programming ('LP') and quadratic programming ('QP') problems, is

however, not relevant to the subject of CUP theory that is the subject of this thesis. LP and QP models depend on a reliance of all the constraints being of a linear form, whereas in CUP theory, as it is formulated here, many constraints that are quadratic or of other, more complex, forms. Chapter 8 will explain that, instead of using the Simplex Algorithm, as used in MPT, CUP theory uses a modified, hybrid form of Monte Carlo simulation. Nevertheless, MPT introduced a conceptual framework that has helped to guide and shape CUP theory and, hence, is explained in this brief chapter.

6.2 MARKOWITZ

In 1952, Harry Markowitz wrote a paper (Markowitz, 1952) entitled “Portfolio Selection” that proved to be seminal. He described a new Modern Portfolio Theory (‘MPT’) that was to give cause for him being awarded the 1990 Nobel Prize of Economics (Markowitz, 1990). This work provided proof of the benefits of diversification.

Prior to the widespread adoption of MPT, the prevailing approach for investors entailed choosing investments that offered the most return for the least amount of risk. Little, if any, regard was given, in any formalised quantitative sense, to the combined effects of such investments. By comparison, Markowitz observed that different investments could be combined - in “portfolios” – such that one could accomplish more than a simple averaging of the constituent investments: that, when combined in a well-formulated combination, the collective return could be more than their average and their combined risk less than their average.

Markowitz described the benefits by highlighting the distinction between systemic and investment-specific risks. His work explained the theory behind the intuition that investors benefit from spreading their trust between multiple investments in different industries. When one industry or share may be under threat of prevailing circumstances, another may be booming... Combined, if the underlying variability in the investments is not positively correlated with each other, or if they are to some extent, negatively correlated to each other, the investor will enjoy a less “bumpy ride” from their combined portfolio, or in other words, a less risky one, than if they “had all their eggs in any one basket”. By adopting MPT, investors can avoid much of the investment-specific risks

even if they cannot avoid the systemic risk that an overall market has in common (across all of the alternative financial instruments).

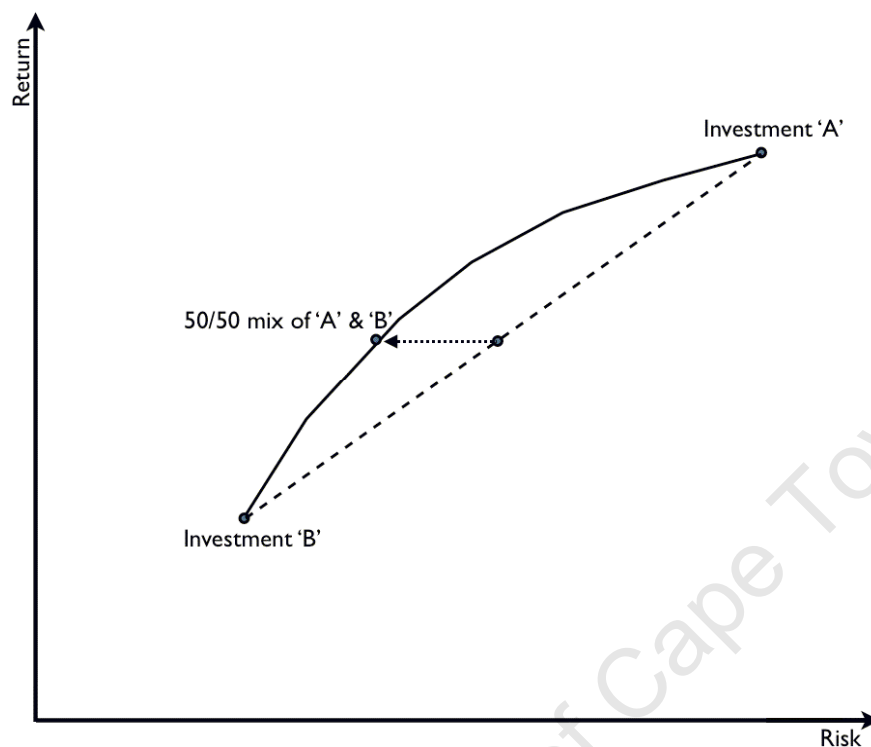


Figure 6.1 Efficient frontier for two alternative investments

Figure 6.1 illustrates the case of combining two unrelated investments. Some portfolio / mix of these two investments (shown on the (always) convex curve that links *A* to *B*) will potentially offer a better trade-off between return and risk than either of these investments if partaken of on their own. One example shown is the 50/50 mix of these two investments. Assuming that the risks that are inherent within both these investments are not the same, that is that they are not exactly correlated to each other, this mix of the two together will attract less risk than if they were exactly correlated (the situation shown by the straight, dashed, line linking *A* to *B*). This illustrates the benefits of diversification. The curved line linking *A* to *B* is known as the *Efficient Frontier*.

When an investor has the choice of multiple investments, their options can be represented on these two axes as a ‘cloud’ of opportunity.

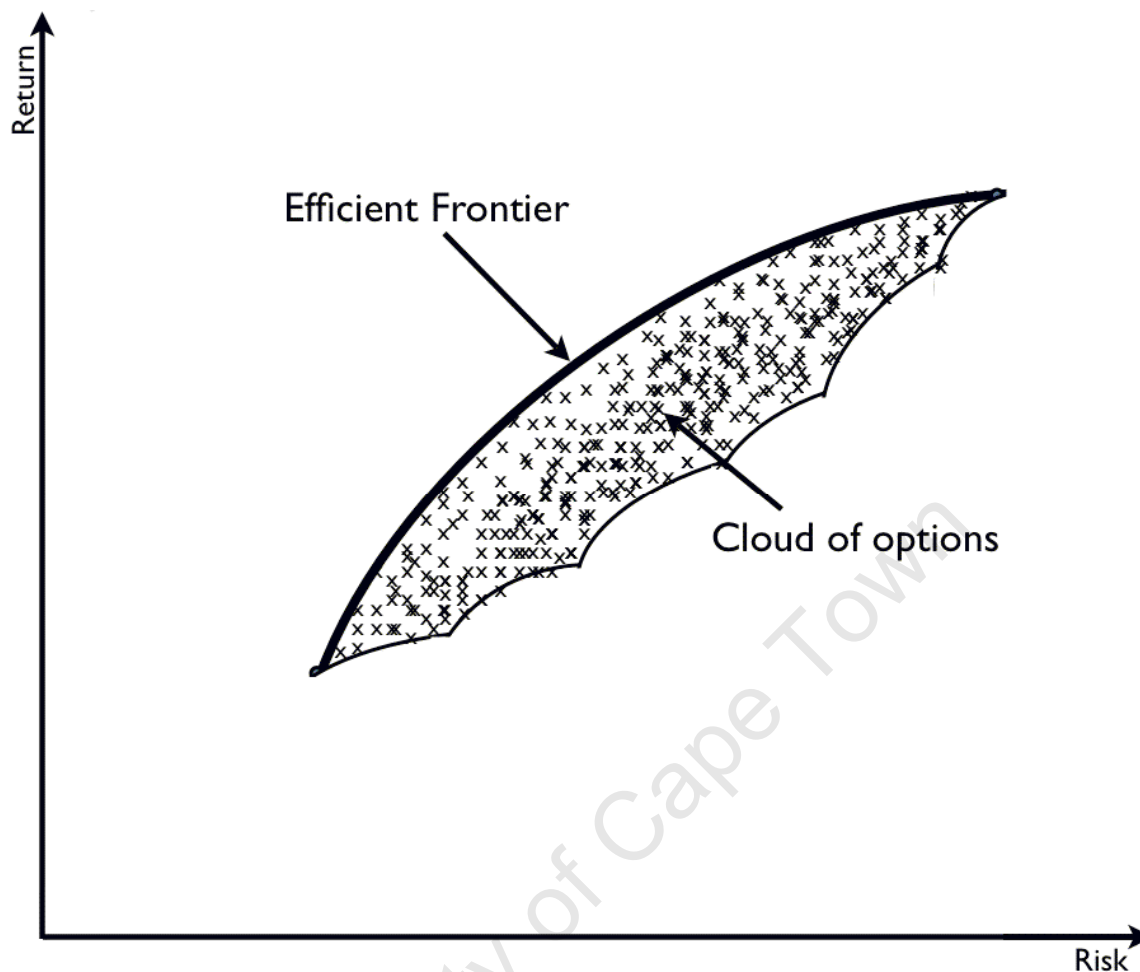


Figure 6.2 The Efficient Frontier of portfolios of investments

Figure 6.2 illustrates this cloud, as well as the *Efficient Frontier*: representing a subset of the prospective portfolios that share a special property: they offer the most return for that amount of risk. Any rational investor could justify choosing any one of the portfolios that lie on this *Efficient Frontier* whereas it would be illogical for them to prefer any of the other possibilities that lie within the adjoining cloud.

6.3 APPLIED TO UNIT PRICING

Item pricing also entails deciding between options, some of which have the allure of greater prospective profits, whilst others have the attraction of less risk. The author has identified that the MPT perspective on this trade-off provides a remarkably useful analysis that can also be applied to unit pricing.

There is, however, an interesting distinction between the domains of investing and that of item pricing. In the latter instance, the contractor has no choice that they have to partake of all the members of the set (that is, that they have no choice that the project comprises the items specified) and they have to do so in the same proportion (i.e. that the portfolio mix between them – in as much as this is determined by way of the item quantities - is a dictated constraint). However, in contrast to investments, the contractor does have choice as regards their pricing of these items. When the contractor is doing their unit pricing, they are also without any control as regards the overall summation of these prices, but they do, nevertheless, have control as regards the nature and extent to which these can be made to be “unbalanced”. By shifting the emphasis around between the constituent components (by assigning them different mark-ups), the contractor can derive greater returns from some combinations than from others; and they can also expose themselves to more risk with some combinations than with others. By contemplating all the alternatives, the contractor can similarly (as do investors) discard any options (such as any low-return, high-risk combinations) that are comparatively “inefficient”, and instead focus their attention only on those that are “efficient”. All of the “efficient” pricing scenarios warrant their attention, and combined they offer a spectrum of choice ranging from low-risk, low-return combinations through to high-return, high-risk alternatives. Their choice of the best one of these depends on their personal attitude to risk and this leads to Chapter 7’s discussion on Cumulative Prospect Theory.

Portfolio Theory has attracted its share of controversy (see, for instance, Bowen, 1984) but has, despite this, become widely accepted into mainstream practice in the field of financial investments, and most especially, equity investments.

6.4 SUMMARY

This chapter has focused only on the conceptual framework that underlies MPT and it has avoided discussion on the mathematics and data used to solve these problems. MPT problems are typically solved as QP models and this nature of model is dependent on the constraints being no more complex than being of a linear form. The item-pricing domain has been found to be more complex, necessitating models that are not as simple as LP or QP models (as adopted by MPT). Nevertheless, MPT provides a conceptual framework,

in particular as regards the *efficient frontier* subset of choices, which is very useful when conceptualizing CUP theory. CUP theory, however, is built using different mathematical tools to those used in MPT.

University of Cape Town

REFERENCES

- Bowen, P.A.** (1984) A hypothesis: portfolio theory is elegant but useless. *The Investment Analysts Journal*, no. 24, November, pp. 17-21.
- Dantzig, G.B.** (1949) Programming in a linear structure. *Econometrica*, vol. 17, pp. 73-74.
- Diekmann, J.E., Mayer, R.H. Jr. and Stark, R.M.** (1982) Coping with uncertainty in unit price contracting, *Journal of the Construction Division, American Society of Civil Engineers*, vol. 108, no. C03, Sep, pp. 379-389.
- Markowitz, H.M.** (1952) Portfolio selection. *Journal of Finance*, vol.7, no.1, pp. 77-91.
- Markowitz, H.M.** (1990) Foundations of portfolio theory. *Nobel lectures in economic sciences 1981-1990*. Ed. Karl-Göran Mäler, World Scientific, pp. 279-288.

7. CUMULATIVE PROSPECT THEORY

7.1 INTRODUCTION

Chapter 6 has provided a hypothesis that the use of Markowitz's Modern Portfolio Theory (Markowitz, 1990) in an unbalanced bidding model can ensure that contractors can sift through all of their pricing options and identify an *efficient frontier* subset of options that are preferable to all others. These options deliver the best possible expected return for their respective levels of risk.

In this chapter, it is advocated that Cumulative Prospect Theory ('CPT') can serve to further reduce the set of *efficient* options to the extent that a contractor can identify, from amongst these, the one pricing combination that will *optimally* suit their risk profile. More specifically, this chapter will trace the origins of CPT through Prospect Theory and Expected Utility Theory to show that it is rooted in Utility Theory.

As with the previous chapter's description of MPT, this chapter serves to describe CPT only to the extent that it is necessary within the context of CUP theory. CPT is a substantial field of research that has attracted its share of controversy. Some of the details as regards this, in particular, as regards the practicality of methods of measuring a contractor's risk profile are certainly relevant here, but have been excluded from the scope of this research. It is later proposed in the Conclusion, that further research in this regard will be useful to further evolve CUP theory in order that it become of greater practical value. It is proposed that the concept of CPT is of use within CUP theory and this has been tested, to some degree, in Chapter 8 as the formulation of CPT has been incorporated into the software that is applied to a hypothetical project.

7.2 AN EFFICIENT FRONTIER

In the previous chapter it was suggested that Modern Portfolio Theory (Markowitz, 1990) can be used to identify the *Efficient Frontier* – representing those pricing combinations

that all offer the maximum expected profit for their respective degree of risk. In other words, they also offer the least possible risk for that amount of expected profit.

Underlying this application of Modern Portfolio Theory to unbalanced bidding is the assumption that contractors are rational – that they resemble the *homo economicus* that is a basic assumption made by behavioral economists (Persky, 1995). In other words, they are averse to taking additional risks without the prospect of an adequate increase in reward. This assumption helps to identify the *efficient* set of item pricing combinations but it does not help with identifying any preference for any single one of these combinations within this set. Some contractors can rationally choose a high-return / high-risk option, whilst others may rationally justify a preference for another choice down at the low-return / low-risk end of the spectrum. Any choice of an *efficient* pricing combination is rational, whereas the choice of any other combination is not.

MPT provides a good illustration that it is logical for contractors to wish to accomplish pricing combinations that will give them the greatest return for the corresponding amount of risk involved. Any such analysis that has this objective will help to greatly focus their attention on a subset of pricing combinations, that could be described as all being *efficient* combinations. This still leaves them, however, with a wide range of choice, covering the full spectrum of degrees of risk to which they could be exposed.

7.3 SUBJECTIVE PERSPECTIVE ON RISK

The concept of risk is one that appears to be difficult to comprehend, not only for contractors, but also for decision-makers in other forms of management. Kahneman and Lovallo (1993) provide an intriguing analysis of typical decision-making by management; concluding that managers' decisions have an irrational bias towards being bold and also overly averse to risk. They argue that decision-makers have a tendency to view their decisions in isolation and with too much of an inside view that is not objectively realistic. They argue that managers tend to be over-optimistic as regards outcomes; for instance, as regards a cost estimate or an estimate of how long an activity will take. This optimism is somewhat compensated for by an irrational tendency to avoid 'sensible' risk-taking, partially caused by a loss of perspective that their job is one of

professional risk-taking. Most managers, apparently, do not perceive themselves to be ‘gamblers’, but rather prefer to think of themselves being able to somehow ‘manage themselves out’ of any exposure to risk (Kahneman and Lovallo, 1993). Kahneman and Lovallo (1993: 17) describe the overall phenomenon as one of “*unjustified optimism and unreasonable risk aversion*”. Within the domain of contracting and, more particularly, that of estimating, this helps explain why estimators might be reluctant to accept an (objective) ‘outside view’ that any estimate they have made is likely to suffer the same problems as previous estimates of theirs have suffered in the past; that they will be as inaccurate as they have been with other projects and that, realistically, a contractor should have little, if any, reason for being uniquely optimistic about their current project.

Economists and psychologists have done considerable research on people’s perspective on risk and on their perceived value of alternatives and how they make decisions relating to these. This field of study falls under what is known as Utility Theory.

7.4 UTILITY THEORY

The field of research in microeconomics regarding people’s perceptions of value is described as *Utility Theory*. The earlier work in this field is well summarized in Fishburn (1989). Utility theory in general entails assessing how people rank the equivalence of various options; such as if they prefer 4 apples to 5 bananas, or 3 apples plus a banana to 1 apple plus 4 bananas, and so on – ignoring any pricing that may be attached to these. These studies then contribute to understanding the relative value (or *utility*) that people attribute to different alternatives and they explain why people make the economic decisions (choices) that they do as regards these goods or services.

Utility theory determines the point at which people become indifferent with regards to alternative choices. Perhaps, for one person, one such situation is that they are indifferent with respect to whether they prefer 3 apples or 4 pears. The *utility* that they consider themselves deriving from either of these options is equivalent in their mind. Notice that these assessments of utility are not *objective* but rather are specific to the person who is involved in making these subjective choices.

These assessments also take account of what is often a diminishing *marginal utility*; that a person may place more value on being given an extra apple when they already have only one such apple, than when they already have (say) 100 of them. The same is true of losses.

7.5 EXPECTED UTILITY THEORY

Many of these assessments entail issues of uncertainty; having to decide on situations where there is either a possibility (but no assurance) of an income or, similarly, the possibility of incurring a loss. These options are often compared against certain alternatives: known then as *certainty equivalents*. For instance, a person may be indifferent as regards whether they are given R70 or have an 80% chance of being given R100. In such a situation, the *certainty equivalent* (of the 80% chance of R100) is said to be R70 even though the *expected value* in this instance is R80. Under such circumstances, it is said that the *risk premium* is 14.3% ($= (R80 - R70) / R70$).

These uncertain situations are widely addressed by *Expected Utility Theory* (see von Neumann and Morgenstern, 1944). This theory gives recognition to people's natural tendency to be either *risk averse* or *risk seeking*. Notice that Utility Theory is based on the subjective *ordinal* assessment of preference; the only significance is that a person ranked one option above, below or as the equivalent of another. It is not popular to attempt to measure an absolute *cardinal utility* (Strotz, 1953) of an option: that is, to try to determine the objective value of utility. Cardinal utility is a far more controversial, and less popular, aspect of economics, as observed by John Hicks (Ahmad, 2001; Chipman, 1995).

By testing people by presenting them with various alternative options, researchers are able to determine their risk profile. Webster (2003) and Besley and Brigham (2007) give examples of this. In essence, a person's or company's risk profile reflects their perception of the acceptable trade-off between risk and the reward required to compensate them for taking this risk. This technique can then be used to predict such persons' *certainty equivalence* of a wider variety of options. With this knowledge one can determine a person's *indifference map* (see Figure 7.1), taking account of different

levels of risk. An indifference map comprises as many as an infinite series of *indifference curves*. Figure 7.1 shows an indifference map comprising three indifference curves - each curve representing the contractor deriving the same level of utility from alternative combinations of expected return and risk. The theory is that a contractor should prefer all options on curve I_3 to all of those depicted on curve I_2 , and so on. However, they should be indifferent as regards choosing any of the options depicted on any one of these curves.

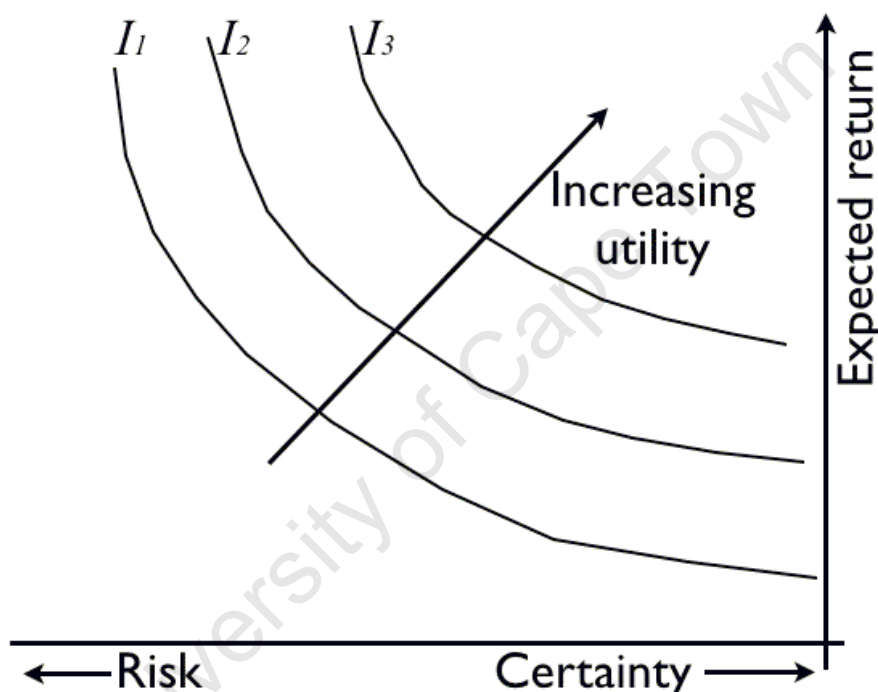


Figure 7.1 Indifference map, comprising 3 indifference curves, each representing a constant level of utility

Figure 7.2 shows the indifference maps for two contractors, A and B . In this instance, contractor B is showing that they are less risk-averse than contractor A . For any given level of risk, contractor A has a greater need for a higher return than contractor B , to compensate them for taking on this degree of risk. Point 'X' shows the risk-free rate of return that both A and B require, shared in common because there is no risk involved. (See Besley and Brigham, 2007.)

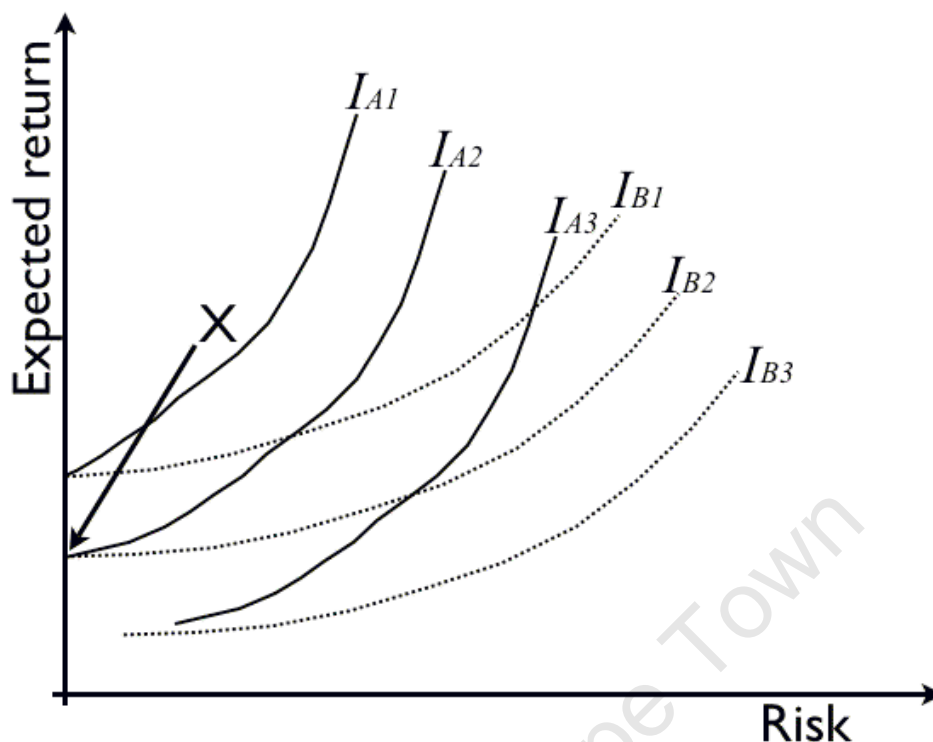


Figure 7.2 Indifference maps for two contractors, *A* and *B*.

If a contractor has had their risk profile assessed and if one then knows their *indifference map*, then the knowledge of this can be combined with that of a project's *efficient frontier* (that MPT has provided). In combination, one can then identify the single (efficient) item pricing combination that represents the contractor's *optimal* choice of pricing for a project, giving them a greater *utility* than all other possible item prices (Besley and Brigham, 2007). This is shown in Figure 7.3, where contractor *A*'s optimal portfolio is depicted as '*M*' whereas contractor *B*'s optimal portfolio is '*N*'. Notice that contractor *B* will not only be deriving more return from their optimal portfolio than contractor *A*, but also that they will be accomplishing greater satisfaction (ie. *utility*) as well, despite incurring greater risk.

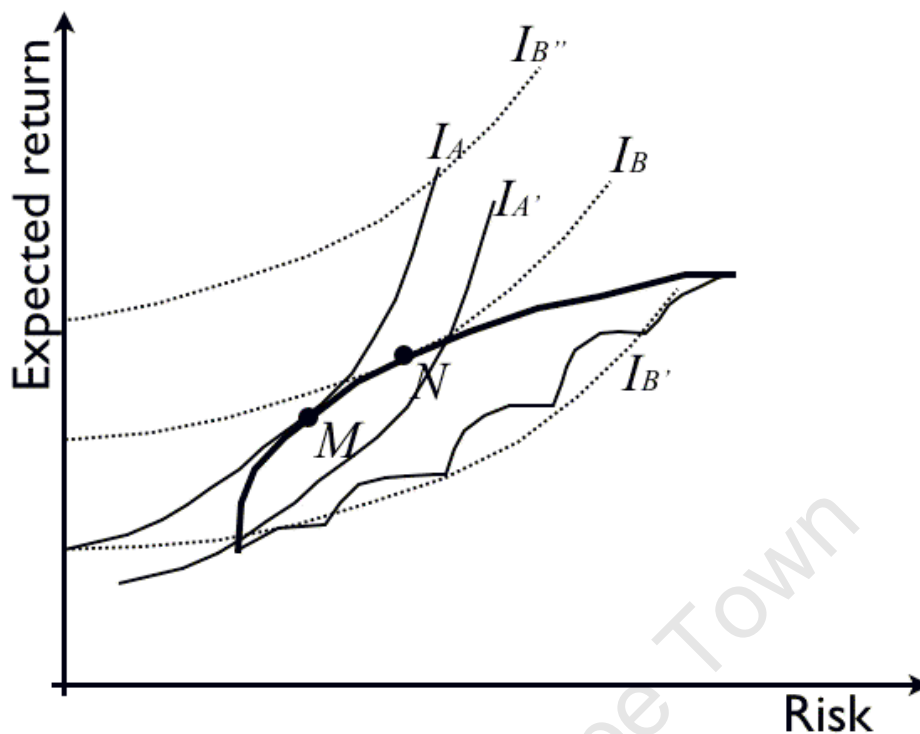


Figure 7.3 Optimal Portfolio Selection

Willenbrock (1973) suggested that bidding models (not referring specifically to *unbalanced* bidding models) could be improved with the use of Expected Utility Theory. Liu *et al.* (2003) have warned though, that in practice, it may be difficult to determine a contractor's utility function for this purpose. The system that has been developed (described in Chapter 8) assumes that this utility function is measurable and that it resembles that which is described by Tversky and Kahneman (1992). However, the practical aspects of soliciting this function needs to be the subject of further research. There has been considerable research done on interviewing and profiling techniques (see, for example, Goldstein *et al.*, 2006), aimed at measuring utility functions, but the approach taken by Willenbrock (1973) has not as yet proven popular within the construction industry and, arguably for this reason, there has been little suggestion so far as regards specifically measuring the risk profiles of contractors.

7.6 PROSPECT THEORY

Expected Utility Theory has been found to deliver some results that are problematic, as illustrated in particular by way of what economists refer to as the *Allais Paradox* (Kahneman and Tversky, 1979) and the *Ellsberg Paradox* (Ellsberg, 1961).

Kahneman and Tversky (1979) proposed an alternative to Expected Utility Theory – termed Prospect Theory (‘PT’), for which Kahneman was awarded a Nobel Prize in Economics in 2002 (Kahneman, 2002). As psychologists, their observation was that people tend to over-react to events with a low associated probability of occurrence and under-react to events that are more probable. Their argument is that people’s perspective on the expected value of events is not in proportion to the probability of the event. They also observed that people’s perspective on possible gains is different to their perspective on possible losses. This concept is illustrated in Figure 7.4.

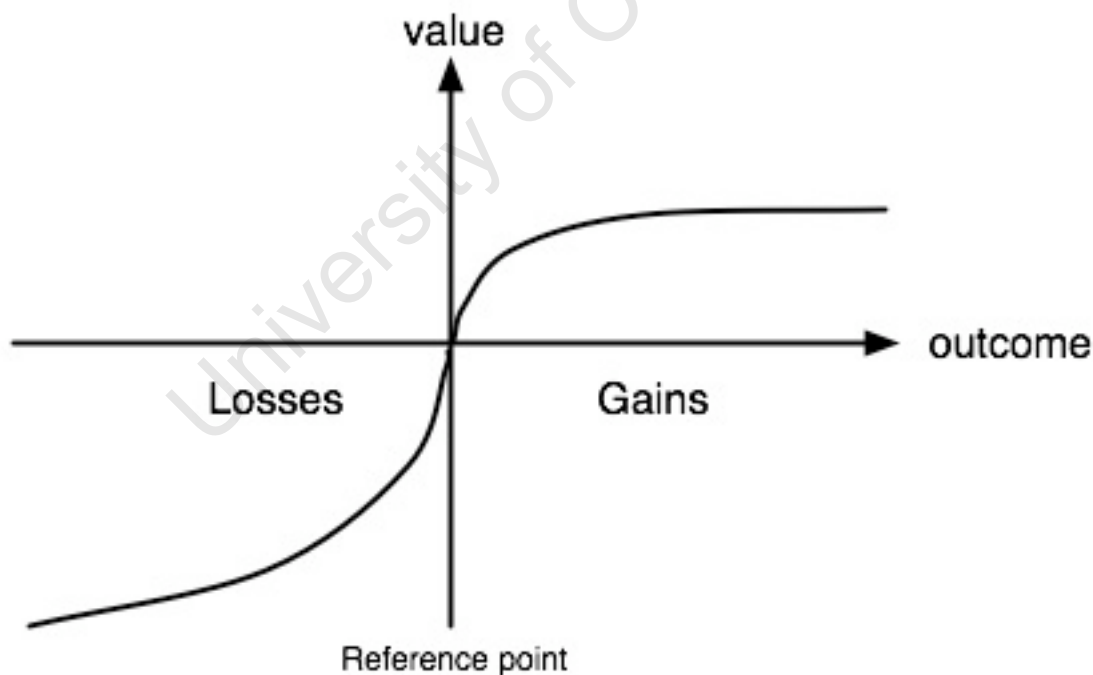


Figure 7.4 Prospect Theory (Kahneman and Tversky, 1979)

Another observation of theirs is that people place no value - and give no consideration to - any aspect that is common to alternatives that is certain to arise. For example, if one

option offers a person a certain R100 plus a 30% chance of an extra R50, and another offers them a certain R50 plus a 50/50 chance of R100, this will be assessed by this person as being the same as comparing the following two options: the first one offering them a certain R50 plus a 30% chance of a further R50, and the other a 50% chance of R100. Both of these alternatives contain a certainty of getting R50 (in the first option, in addition to another certain R50) and so, their research shows, that people take this R50 for granted and they discard this from having any influence on their decision between these two options. People have been found to only give consideration to those things that are different between their choices.

It is argued that Prospect Theory serves better (than Expected Utility Theory) to explain why many people enjoy the risk of a lottery ticket and yet will often, at the same time, insure themselves against losses. In other words, many people consciously seek some risks (such as the extreme risks of paying for a lottery ticket with almost-certain assurance of loss) and yet they simultaneously justify the payment of an insurance premium to give them the peace-of-mind of protection from far-less likely risks such as those that are insurable. People's attitude to the uncertainty of gains is different to their attitude as regards the uncertainty of losses. Their findings are that people are generally *risk averse* when it comes to gains but *risk seeking* when it comes to losses. This is illustrated in Figure 7.4 by way of the concave shape for gains and the convex shape for losses. When weighing up the choices between alternative gains, people tend to be risk averse and place a higher proportional weight on options that are more assured. When it comes to losses, the opposite is true; they observed that people tend to apply a lesser proportional weight on losses that are more likely and irrationally over-emphasise their fear of highly-unlikely losses. They found that the utility curve for most people is steeper for losses than it is for gains, explaining that people have an irrational tendency to be less willing to gamble with profits than with losses (Tvede, 1999).

7.7 CUMULATIVE PROSPECT THEORY

Prospect Theory has attracted considerable further research since its initial formulation in 1979. Tversky and Kahneman (1992) themselves identified and published an improvement to their original theory that is called *Cumulative Prospect Theory* ('CPT'). This theory is also derived from concepts inherent within *rank-dependent expected utility theory* ('RDEU') (Quiggin, 1982).

This improved theory (Tversky and Kahneman, 1992) identified a four-fold nature of behavior: risk aversion for gains and risk seeking for losses of high probability; and risk seeking for gains and risk aversion for losses of low probability. Such behaviour is taken account of by way of incorporating an additional weighting factor that is graphically shown to resemble Figure 7.5. Different people act on the basis that they prefer different weights and it is this difference that determines their *risk profile* – that is, that describes their attitude to different degrees of risk. An analysis of the differences between CPT and PT is provided by Fennema and Wakker (1997).

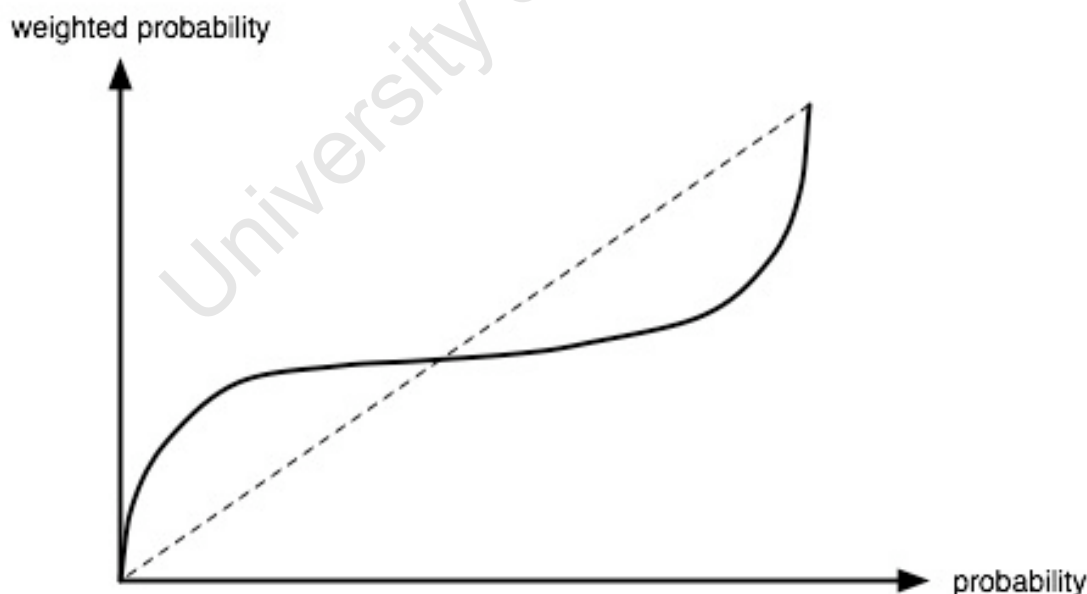


Figure 7.5 Cumulative Prospect Theory weighting factor (Tversky and Kahneman, 1992)

Individual contractors can be interviewed and presented with a series of choices that involve varying degrees of risk. Ideally, these questions will relate to the specific domain of item pricing, rather than be of an abstract form such as if the questions were to relate to various choices of lottery tickets. As Stott (2006) has warned, one must not expect the contractor to always be consistent in any such interview. Tversky and Kahneman (1992), amongst others, have sought to fit various parametric-based functions to describe subjects' risk profiles. Abdellaoui *et al.* (2007) have instead proposed a method by which to elicit a parametric-free utility function and Stott (2006) has confirmed substantial advantages of a parametric-free approach. More specifically, this method entails the elicitation of *utility midpoints* - where the subject is indifferent between two alternative choices. Abdellaoui *et al.* (2007) noted that they consider their method very efficient at being able to elicit a subject's utility function across the whole domain, as a derivation from few measurements – therefore entailing a realistically efficient interview in each instance.

Regardless of the technique of elicitation, CPT identifies the utility $V(g)$ that an individual will attribute to a circumstance where they are presented with an n -outcome uncertain opportunity $g = (p_1, x_1; \dots; p_n, n_n)$ where p_i is the probability of x_i with the x_i 's ranked in descending order of value. Bear in mind that CPT differs from *Prospect Theory* by way of the use of a series of additional weighting factors w_i where $w_1 = p(p_1)$ and

$$p(p) = \frac{p^r}{(p^r + (1-p)^r)^{\left(\frac{1}{r}\right)}}$$

and $w_i = p(p_1 + \dots + p_i) - p(p_1 + \dots + p_{i-1})$ for $1 < i < n$

where $p(p)$ is a monotonic risky weighting function constrained by

$$0 \leq p(p) \leq 1$$

Tversky and Kahneman (1992) suggested a parameter setting of 0.61 for r given the prospect of a variety of gains, and 0.69 given the prospect of losses. These values have been implemented in the software that has been developed and that is described in Chapter 8. In practice, this would need to be customised to each individual contractor, based on an interview, to take account of their unique risk profile.

Stott (2006) observed that people's risk-taking decision-making behaviour is typically stochastic, rather than deterministic. Stott found that people change their minds and do not consistently make the same decisions, given the same choices. To cater for this, Stott advocates the addition of a stochastic choice function to the deterministic CPT function, in order that it will perform in a stochastic manner. Loomes and Sugden (1995) also suggest that this nature of adjustment should be regarded as an integral part of CPT. This has, however, not been taken account of in the software that is described in Chapter 8 and this idea gives potential for further research.

7.8 PRACTICAL APPLICATION OF THE THEORY

Keeney and Raiffa (1993) describe the process by which one can apply a utility function in practice, such as one derived using Cumulative Prospect Theory, involving two attributes: the most likely return as well as the Value-at-Risk. In essence, one is seeking to determine the contractor's preference to alternatives that involve different degrees of each of these factors. The goal is to identify the *value* (the term used by Kahneman and Tversky (1979), as opposed to the term *utility* that is used by von Neumann and Morgenstern (1944)) that the contractor will derive from a particular item pricing combination, given that it is estimated that this combination will generate a given expected return subject to an assessed Value-at-Risk.

The software application of CUP theory, which is described in Chapter 8, incorporates the formulation of the *value* as identified by Tversky and Kahneman (1992). This has not been modified or adjusted as would be required to describe the unique personality, in terms of their risk profile, of a particular contractor, as could be assessed by way of the interview processes referred to above. Issues as regards this personalisation process have

not been researched here seeing as it has been deemed to be beyond the scope of this thesis, and this area provides opportunity for further research.

7.9 SUMMARY

Considering that a contractor has the choice of a considerable number of different pricing combinations when deciding their item pricing, Chapter 6 had proposed a basis by which a contractor might be able to narrow these down to a small subset that share the quality of being *efficient*. *Efficiency*, in this sense, refers to the characteristic of offering the greatest amount of return for that degree of exposure to risk, or in other words, the least exposure to risk for that amount of return. This would help considerably to sift out all of the many pricing options that are inherently not of a nature that one would rationally wish to consider them, but this still leaves a spectrum of choice. This spectrum would span options providing low returns with low risk, all the way through to some that would generate high returns along with high-risk - all of which could be considered attractive, depending on one's personal appetite for risk.

The research has therefore led to consider the mainstream microeconomic techniques for situations that involve personal assessments of risk, and this has given rise to consideration of Cumulative Prospect Theory as offering one of the most widely accepted methods of addressing this.

This chapter has provided a framework, making use of Cumulative Prospect Theory, by which a contractor could identify their *optimum* item pricing taking into account the effects of item prices on both the expected return that a contractor will enjoy as well as their risks. This model takes account of the contractor's appetite for risk and finds the solution that they will find 'most satisfying' given recognition of their psychology as regards tackling projects with various prospects of returns and risks.

Having outlined a framework to identify the one choice that offers the best-suited compromise, particular to an individual's unique attitude towards risk, the next chapter describes testing CUP theory and its associated composite model by way of software that

has been written especially for this purpose. This is then applied to the data of a hypothetical project to see the results.

University of Cape Town

REFERENCES

- Abdellaoui, M., Blauchrodt, H. and Paraschiv, C.** (2007) Loss aversion under prospect theory: a parameter-free measurement. *Management Science*, vol. 53, no. 10, pp. 1659-1674.
- Ahmad, S.** (2001) Hicks on capital. *John Hicks: his contributions to economic theory & application*. Ed. K. Puttaswamaiah, London: Transaction Publishers, pp. 43-60.
- Besley, S. and Brigham, E.F.** (2007) *Essentials of managerial finance*. 14th.ed., Cengage South-Western. ISBN: 0-324-42270-9.
- Chipman, J.S.** (1995) Hicksian welfare economics. *The legacy of Hicks*, Eds. H. Hagemann and O.F. Hamouda, London: Routledge, pp. 93-142.
- Ellsberg, D.** (1961) Risk, ambiguity, and the Savage axioms. *Quarterly Journal of Economics*, vol. 75, no. 4, pp. 643–669.
- Fennema, H. and Wakker, P.** (1997) Original and cumulative prospect theory: a discussion of empirical differences. *Journal of Behavioral Decision Making*, Vol. 10, pp. 53-64. Available at <http://people.few.eur.nl/wakker/pdfspubld/97.3optcptjbdm.pdf> [accessed 12th May, 2009].
- Fishburn, P.C.** (1989) Retrospective on the utility theory of von Neumann and Morgenstern. *Journal of Risk and Uncertainty*, vol. 2, pp. 127-158. Available at <http://www.springerlink.com/content/g0h4518264m80t76/fulltext.pdf> [accessed 12th May, 2009].
- Goldstein, D.G., Johnson, E.J. and Sharpe, W.F.** (2006) Measuring consumer risk-return tradeoffs. SSRN Working Paper. Available at <http://ssrn.com/abstract=819065> [accessed 5th June, 2009].
- Kahneman, D.** (2002) Maps of bounded rationality. *Prize Lecture*. The Nobel Foundation. Available at http://nobelprize.org/nobel_prizes/economics/laureates/2002/kahneman-lecture.html [accessed 12th May, 2009].

- Kahneman, D. and Tversky, A.** (1979) Prospect theory: an analysis of decision under risk. *Econometrica*, XLVII, pp. 263-291.
- Kahneman, D. and Lovallo, D.** (1993) Timing choices and bold forecasts: a cognitive perspective on risk taking. *Management Science*, vol.39, no.1, pp. 17-31.
Available at
http://www.som.yale.edu/faculty/keith.chen/negot.%20papers/KahnemanLovallo_ChoicForecastsRisk93.pdf [accessed 12th May, 2009].
- Keeney, R.L. and Raiffa, H.** (1993) *Decisions with multiple objectives: preferences and value tradeoffs*, 2nd ed., New York: Cambridge University Press.
- Loomes, G. and Sugden, R.** (1995) Incorporating a stochastic element into decision theories. *European Economic Review*, vol. 39, pp. 641-648.
- Liu, S., Lai, K.K. and Wang, S.** (2003) Several mathematical models for competitive bidding and auctions. *Advances in Management Science Studies in China*, vol.1.
Available at <http://site.uibe.edu.cn/JMTMSUI/UploadFiles/Files/lsl2.pdf>
[accessed 12th May, 2009].
- Markowitz, H.M.** (1990) Foundations of portfolio theory. *Nobel lectures in economic sciences 1981-1990*. Ed. Karl-Göran Mäler, World Scientific, pp. 279-288.
- Persky, J.** (1995) Retrospectives: the ethology of Homo Economicus. *The Journal of Economic Perspectives*, vol. 9, no. 2, pp. 221-231.
- Quiggin, J.** (1982) A theory of anticipated utility. *Journal of Economic Behavior & Organization*, vol.3, pp. 323-343.
- Stott, H.P.** (2006) Cumulative prospect theory's functional menagerie. *Journal of Risk and Uncertainty*, vol.32, no.2, pp. 101-130. Available at
<http://www.springerlink.com/content/a48206wm08n656u6/fulltext.pdf> [accessed 12th May, 2009].
- Strotz, R.H.** (1953) Cardinal Utility. *American Economic Review*, vol. 43, pp.384-397.
- Tvede, L.** (1999) *The psychology of finance*. New York: Wiley.

- Tversky, A. and Kahneman, D.** (1992) Advances in prospect theory: cumulative representation of uncertainty. *Journal of Risk and Uncertainty*, vol. 5, pp.297–323.
- von Neumann, J. and Morgenstern, O.** (1944) *Theory of games and economic behavior*. Princeton: Princeton University Press. Section 3, chapter I. Reprinted in *Utility theory: a book of readings*, ed. A.N. Page (1968), New York: Wiley, pp. 215–233.
- Webster, T.J.** (2003) *Managerial economics: theory and practice*. New York: Academic Press.
- Willenbrock, J.H.** (1973) Utility Function Determination for Bidding Models. *Journal of the Construction Division, American Society of Civil Engineers*. vol. 99, no. 1, July 1973, pp. 133-153.

8. SOFTWARE FOR TESTING THE MODEL

8.1 INTRODUCTION

Chapter 4 described the three benefits that contractors can derive from item pricing (namely, improvements in cashflow, variation valuations and the value of escalation compensation) and it provided a model by which to quantify these, given any item price combination.

Similarly, Chapter 5 defined the three types of risk that are produced by item pricing (namely, the risks of *rejection*, *reaction*, and of *being wrong*) and, again, a model was provided to quantify these on a collective basis.

Chapter 6 then suggested that MPT has provided a framework of understanding that facilitates that a contractor could logically discard most pricing combinations, if given knowledge of the corresponding returns and risks that these will generate, thereby facilitating that a contractor can focus their attention on a small subset of *efficient* pricing combinations. *Efficiency*, in this context, is defined (by way of MPT) as the greatest return to be had at any defined degree of risk.

This analysis would still leave a contractor with a spectrum of choice, ranging from some pricing that gives high returns with high risk, through to pricing that has far less risk but only offers low returns. It is at this point that a contractor would need to assess their appetite for risk and it was suggested, in Chapter 7, that CPT provides a popular basis by which to make this nature of assessment. These assessments lead to an ability to determine a *value of utility* for any given pricing combination, with the opportunity then to seek the one combination that will deliver the most *value*. Given this theory, the 'revenue' and 'risk' models from Chapters 4 and 5 can be combined as a composite model by which to determine both the return and risk, and hence also the *value of utility*, generated by any given item price combination.

The current chapter describes how this composite model is now encoded in software, using Monte Carlo simulation, guided by a novel hybrid use of a range of techniques that could be described as artificial intelligence, genetic computing and fuzzy logic. This system is able to identify the *quasi-optimum* item pricing scenario for a project.

The full source code for this system is listed in Appendix A. The next chapter describes the results of a test, using data that describes a small hypothetical project (provided as Appendix B). Appendices C, D and E provide details of the test results and the pricing schedule that has been identified by the model as providing the highest value for the contractor in the instance of this project.

8.2 THE JAVA TECHNOLOGY PLATFORM

The software has been written in Java (see online at <http://java.com>), which is a popular, sophisticated modern programming language, not dissimilar as regards its syntax to C and C++, on which it is largely based. Java is regarded as being part of the ‘C family’ of programming languages and Microsoft’s proprietary language that is very similar to Java is even called C# (pronounced C Sharp). Java was developed by Sun Microsystems, and released by them in 1995, and is now published as ‘open source’.

Java is a language that is ‘platform-independent’ in the sense that programs written in Java will run on any hardware that supports a Java Virtual Machine (‘JVM’). JVMs are available for a vast range of hardware, spanning from mobile phones to supercomputers. Java was also intended for embedded systems and is a popular choice for the control systems of appliances such as domestic microwave ovens and heat pumps. By comparison, other languages (such as Visual Basic (‘VB’), C, C++ and C#) are instead written and compiled to run on a specific choice of hardware platform.

The tests done here were run on an Apple MacBook Pro laptop computer (with a 2 GHz Intel Core Duo processor, with 2GB of RAM), running Apple’s OS X operating system, but the same program could be run on Windows or Linux-based machines, ranging from laptops to mainframes. Even supercomputers, at the top-end of the spectrum of hardware

that is available, can inevitably run Java programs and could easily run the program listed in Appendix A.

The program has been written for the specific purpose of providing this test as part of this thesis. It has not been written with the intention of serving as ‘commercial code’ as would be required to be suitable for an industrial ‘real-world’ deployment. It is not written so as to be robust enough for this purpose, nor as fast, as easy to use, as adaptable or as easy to maintain as would be required in a commercial environment. These factors have been deemed immaterial for the purposes of the academic exercise of this thesis.

This program is highly process-intensive and it takes around a week to run on the laptop computer described. This is obviously not a concern for the purposes of the academic exercise involved here, but would be an issue commercially. There are, however, many ways that this source code could be adapted so that the program will run very much faster than it does now, by undertaking a process of evolutionary improvement known as ‘code optimisation’. It could also be made faster by simply deploying it on faster hardware and it is of a nature that it very much lends itself to easily being able to be adapted to run on a massively parallel-processing (‘PP’) system. The nature of the software is such that this PP system could be fairly easily deployed on a network of computers, each assigned to run one or several ‘threads’ of the overall process, with these reporting back to a central server on their progress, thereby being centrally controlled and co-ordinated by way of settings issued to each of these threads as a guide to ensure that their processing results will contribute positively to the overall effort.

The software implements the models which analyse both the expected rewards as well as the risks that are generated from a range of different item prices and then combines these to measure the contractor’s derived value of utility, using Cumulative Prospect Theory (see Chapter 7). This software system uses Monte Carlo Simulation (‘MCS’) (see Section 8.3.1 below) combined with some regular intervention and guidance derived from fuzzy logic (see Section 8.3.4 below). It runs until it is no longer improving upon its previously found best solution and until it determines that there is no longer a reasonable prospect of any further such improvement. Alternatively, it terminates when

two million iterations have been run if the system has not found adequate cause to decide to terminate prior to this. In practice, when testing the hypothetical project, it terminates after about one million iterations. The two million iteration limit is implemented simply to serve as a scope for the declaration of some arrays used for the recording (for subsequent reporting) of the progress of this system.

8.3 OVERVIEW

Consideration should first be given to a broad overview of the manner in which the system has been coded, before later delving down into the details. The overall objective of the software is to find the item price schedule for the project that produces the greatest *value of utility* for the contractor. In broad principle, this schedule needs to comprise a choice of price for each item that, if considered in isolation, will deliver a high contribution of *value of utility*; as well as an optimum combination of these prices.

The software is designed to incorporate a combination of Monte Carlo simulation, artificial intelligence, genetic computing, and fuzzy logic.

8.3.1 Monte Carlo simulation (MCS)

MCS describes a nature of approach to computing that entails using many random samples of data as a simulation of typically large, complex systems. It is an approach widely adopted in complex financial economic analyses, and as well as in the physical sciences, mathematical sciences, and other areas of science (see Rubinstein and Kroese, 2007). It is typically used in situations where the complexity is so great that it is impractical or inconvenient to apply any alternative mathematical approach.

This approach is well suited when a complex model can be broken down into components that can each, individually, be modelled, and yet where the combination of all these parts though is such that a parametric formulation of the overall problem becomes too complex.

The use of MCS facilitates that one can dissect the overall domain into constituent elements, model each of them and then combine them in a 'random' fashion which

nevertheless respects the observed interaction between these elements. It can be made to *simulate* the overall complexity very effectively.

In financial economics, it is especially well suited to issues that are, essentially, inherently of a stochastic nature, as opposed to being of a deterministic nature (see Glasserman, 2003).

MCS is said (Metropolis, 1987) to have been developed in the Los Alamos National Laboratory in the 1940s, for the purposes of research, by the government of the United States of America, into nuclear armaments. It was first applied to the field of finance in 1964 (Hertz, 1964) and has become increasingly popular due to computer processing power having grown exponentially. This computing power has also become increasingly inexpensive and accessible (see Moore, 1965).

8.3.2 Artificial intelligence (AI)

Artificial intelligence, a term first used in 1956 by John McCarthy (Crevier, 1993), refers to a branch of computer science that has the purpose of making ‘machines’ (i.e. computers) operate with some ‘intelligence’. The concept is that, rather than a computer operate entirely under direct instruction, it can be set up (i.e. programmed) so that it may have its own awareness of its environment; that it can have an accumulated knowledge (and continue to augment this, by itself, with further learning); and reason and decide for itself how it should respond to its environment or any given situation. The programmer, in such instances, has to ‘*step back*’ and rather than program each response that the system needs to make when given any input, they need to equip the system with a ‘set of tools’ – a logic – that will facilitate that the system is able to make its own assessment of any given ‘input’, so that it can decide for itself how it needs to respond. Programming in AI entails a programmer equipping a computer with a means to think for itself and hence become, to a degree, self-sufficient. AI systems can be thought of as having their own personality and psychology.

Some programming languages, such as LISP or Prolog, have been specifically designed so as to be well suited to AI. However, some degree of AI functionality can be built into any system, and written in any programming language, such as Java (Watson, 2002).

8.3.3 Genetic computing (GP)

GP is a type of artificial intelligence where a system is made to evolve in a style similar to biological evolution (Banzhaf *et al.*, 1998). Such systems seek to improve themselves by selectively discarding some of their knowledge and characteristics (or options), discounting these from proceeding to have any further influence on the system as it goes forward, whilst mutating other aspects of itself. It is a new field of computer science that is, as yet, ill-defined (Banzhaf *et al.*, 1998) and the program that has been developed here is arguably of a form that fits this definition. Nevertheless, the system in question very much has *evolutionary* characteristics, it certainly does selectively discard options and choose its 'genetic' make-up to take forward for further processing, and it is designed to run in phases (or 'waves') that have, for the purpose of acknowledging these GP-type characteristics, been called '*generations*'. The results from each generation are analysed for the purpose of determining the nature and direction of the generations that succeed it. The system accumulates a knowledge that each generation passes on to the next one, whilst helping set up the next generation so that it might perform better than those that preceded it. Each generation becomes more intelligent than its ancestors. For more on GP, see Holland (1992), and Davis and Mitchell (1991).

8.3.4 Fuzzy logic

Fuzzy logic (Zadeh, 1965) is a technique used, inter alia, in the field of artificial intelligence to handle situations where there is imprecision and uncertainty. It is suited to modeling domains that are too complex to describe with deterministic, parametric precision, but nevertheless where the characteristics of the domain are well known, albeit with a degree of some 'vagueness' (see Novak *et al.*, 1999).

Systems using fuzzy logic have become very common in the past decade, to the extent that our modern daily living is surrounded by their influence. They are popularly embedded into domestic household appliances, such as air-conditioners, heaters, washing

machines, dishwashers, tumble-dryers and microwaves. They are very well suited to provide effective control of situations such as determining whether a tumble-dryer would be more efficient running faster or for longer or with more heat, or should stop - given some vague nature of input (such as a combined 'knowledge' of the humidity and weight of a load of washing, given a user's objective and description of the nature of the load). Such situations may be difficult – perhaps, at times, for reason of the vast number of possible alternatives to choose between – to model with absolute mathematical precision. Nevertheless, given repeated reassessment in real-time by a system that incorporates fuzzy logic, and the processing required becomes relatively minimal, very reliable and effective.

The software listed in Appendix A incorporates a style of fuzzy logic in as much as it has been designed not to be dependent on the input of exact data, nor on having to have exactly determined functions by which all steps in the process are calculated. For instance, it is designed to monitor the results that it accomplishes with certain settings and if these results are not proving to be of value as regards contributing to the knowledgebase, it adjusts these settings (by some degree that is not having to be precise) and tries again. It's own inbuilt logic then leads it to keep trying other settings, depending on how well it is performing as regards improvements on the knowledgebase. This process is guided by an imprecise, fuzzy logic and yet when knowledge is found that is of value as an addition to the knowledgebase, this has not suffered in quality, as regards its validity, for reason of any vagueness in the process that led to this discovery.

8.3.5 Hybrid

The software developed employs a hybrid of the above techniques. It uses AI, GP and Fuzzy Logic to guide a process of MCS. MCS could be employed as the sole approach but this would be a very time-consuming procedure, having to use a very large number of tests if it were to prove effective. However, as a rough guide to the number of iterations that would be required, let us assume that there are in the order of 1000 items in a project and that we would like to explore a price range for each of them that is, on average, a range of around R20.00 with a granular accuracy of 1c. On this basis, there would be 2000 possible alternative prices for each of the 1000 items and the goal would be to

identify the best performing combination of these. If one were to want to test them all, this would amount to 2000^{1000} tests ($= 1.07E3301$) which is an impractically large problem to compute within a realistic timeframe, especially given that this analysis will furthermore need to test stochastic data to take account of variability in the various risk factors.

Instead, logic can be introduced that is founded on the knowledge of the theory as regards unbalanced bidding, as it has been presented in the preceding chapters. This logic is encapsulated in a combination of AI, GP and Fuzzy Logic – and this serves to considerably reduce the amount of testing that is required. This logic guides the MCS with the effect that far fewer tests (i.e. iterations) are required before the optimum (or *quasi*-optimum) pricing combination is identified.

The overall objective is to identify the item price combination that will deliver the highest possible *value* to the contractor. This *value* assessment takes account of the contractor's identified *risk profile* as regards their attitude towards alternative propositions of reward vs. risk – as discussed in Chapter 7. Thus, in broad principle, the program needs to assess the reward that is generated by way of a possible item price combination, as well as the risks that would be attributed to this. By way of this, the system is then able to assess the *value of utility* generated from each pricing regimen and then compare these so as to identify the one that produces the highest *value* for the contractor. In essence, this winning pricing combination will give the contractor the best combination of a relatively high profit together with a comparatively low risk (measured in accordance with the contractor's perception of risk).

The challenge lies with finding a way to reduce the number of tests required to an extent that this becomes practical in terms of the time required to process these. The AI chosen to facilitate this requires that it have a *knowledge* of the domain which, when assessed by way of its inbuilt logic, determines that it will be 'safe' to exclude a proportion of all further tests for reason that these are unlikely to produce a competitive *value* by comparison to other alternatives. This knowledge or *intelligence* is accumulated en-route as the system processes the domain and gets to understand its dynamics better and better.

This knowledge is acquired at two levels, as follows:

1. Knowledge of each item, taken in isolation. This level of the analysis discards (/ ignores) any consideration of the role that any price assigned to this item plays within the broader context of the overall project.
2. Knowledge of the effect on the *value* that will be derived from the overall project, from various combinations of unit prices being assigned to the project's constituent items. Keep in mind that, overall, the item prices are constrained by having to summate to the overall tender price that the contractor has decided for the project. Any price that is assigned to any one item can be viewed as effectively amounting to a forfeiture of the prospects of assigning these same funds to another item. In this context, each item can be viewed as being competitive with all the other items in the sense that whatever pricing is assigned to one item, this amounts to a reduction in what can then be assigned to other items.

The best pricing for a project does not simply comprise a combination of the best pricing for each item, assuming that the latter was determined without consideration of the project as a whole.

If any one item is taken in isolation, the best unit price for this item, if one were to ignore risk, would be the full tender price divided by the quantity of the item, i.e. the maximum possible price. The higher the price, the more profit the contractor will derive. However, it has already been shown that item pricing needs to be moderated by the risks involved and the risks of very high prices will outweigh the benefits that these generate by way of profits. Patently, it would be ridiculous if the contractor were only to give a price to one item in a whole project, assigning this one item its maximum possible price. Such pricing would inevitably be treated by any client as an unacceptably unbalanced bid, and would, almost certainly, be rejected. Thus, if one is to consider a price relative to its level if the pricing were decided on a balanced basis, there might be thought to be a *centrifugal* force, motivated by a pursuit of profits, that drives pricing up; whilst there is a *centripetal* force, motivated by the avoidance of risk, that drives the pricing back towards

its central / balanced state. These two forces counterbalance each other, depending on each individual contractor's perception of risk, settling at some point where they consider themselves to be getting a suitable and ample reward for taking on the risk that is involved (at that price-point). Any higher price would be considered to be generating too much risk for this amount of profit, whilst any lower price would be thought to be generating insufficient profits, even though the risk is less.

On this basis, if taken in isolation, one could identify the optimum pricing for each item. However, the aggregation of these optimum item prices is highly unlikely to equate to the tender price to which the contractor is already committed.

Taking an example of a situation where a process of analysis has identified that the optimum pricing for a project's items, if each is assessed individually in isolation, summates to an amount that is (say) 20% more than the overall tender price. Further analysis would be required by which to compromise on these previously-found 'optimum' prices, such that they will work together in unison to produce the best *overall* utility for the contractor.

The best compromise in such a situation is likely to comprise some items retaining their 'optimum' prices (that were found to be optimum when taken in isolation) whilst other items will end up having their prices reduced, to varying degrees, such that together they will satisfy the constraint by which they all summate to the tender price.

This system has the end-goal of finding the optimum combination and en-route it has the objective of not only constantly trying to reach this goal (before too much processing time has elapsed), but also to building up its knowledge of the domain – equipping it so that it will become more and more knowledgeable, as it proceeds, and thereby better able to pursue this goal at a later stage in the process. It presumes that the more knowledgeable it becomes, the more likely it will be that it will more quickly reach its goal.

The system has been given the 'attitude' that it can embark, at some stages, to pursue further knowledge of the domain, where there is very little prospect of any chance that it

might, in the short terms, ‘stumble upon’ (for reason of its random endeavours) a pricing combination that is likely to compare favourably with those that it has found to be the best-suited thus far. These phases are motivated purely because of a recognition that by investing in the further accumulation of this improved knowledge, it will have gained something that will serve as an asset. It recognizes that by investing in its own ‘self-improvement’ it will, at times, accomplish more than if it were to stick to the ‘straight and narrow’ – focusing in solely on pricing that it has thus far found to be most productive in producing the best overall results. In the process, by taking a chance testing less obvious pricing options, it might then become able to reassess its prior ‘beliefs’ as regards the price points at which best results are to be found.

For example, let us say that early tests indicate that an item is producing its best *value of utility* at around a price level of R100. The system’s knowledge, at this early stage, might indicate that prices higher than this produce inadequate risk to compensate for the higher profitability, whereas lower prices simply do not produce attractive-enough profits. The system could, at such a stage, decide to focus in solely on prices very close to R100.00 with the idea being that it can stop wasting processing time testing any other level of price in combination with all the other item prices. The system could be making similar decisions at the same time, as regards each of the other items. However, early indications may be wrong and the system recognizes that, perhaps, in this example, better pricing might ultimately be discovered at around (say) R120.00. To accommodate this, the system, in effect, records that the best pricing thus far, for this item, is obtained at around R100.00 but at the same time, it indulges its desire for better. It acknowledges the prospect of a better-performing price and so it proceeds, going forward, at times testing to see the effect of prices of around R100.00, in combination with all the other items; and at other times, scouting out and seeing if other prices might do even better. This nature of the system gives cause for many discoveries and so, if for instance, the system finds that prices of around R120.00 become even better than prices of around R100.00, it begins to use pricing at around both these price-points for a period, until it might become convinced that pricing around R120.00 is almost always better than pricing the item at around R100.00. It then fully acknowledges this new intelligence and allows the pricing to gravitate more towards the use of R120.00 than R100.00 for this item.

The system is designed therefore to ‘go off on tangents’ – to, at times, explore new territory distant from where it would seem sensible to be focused, recognizing that the overall pricing exercise is incredibly complex and that new discoveries may be made that no parametric-driven algorithm could suggest. A good example is where the system is encouraged to ‘let go’ of trying too hard to find the best prices and instead, simply, rather treat as more important, the accumulation of a better understanding of the problem at hand. By doing so, it expands upon its knowledgebase, getting a better understanding that (hopefully) leads to new discoveries and, ultimately, to improved pricing.

Besides the pursuit of pricing that produces the best utility for each item, the system does similarly as regards combinations of item prices that produce the best utility for the project as a whole. It pursues the hypothesis that the best pricing combination will contain many, but not all, of the best prices for each item (produced in the event that the item is isolated from the added constraints of being a constituent of the whole project). Working on this basis, the system is designed to seek knowledge of good prices for each item, in isolation, as well as good combinations of prices, as a collective whole.

8.4 THE PROGRAM

The source code, written in the Java programming language, is attached as Appendix A. The program is written so that it reads the data (describing the project, attached as Appendix B) from a file on disk, processes it, reports on its progress (an example of which is attached as Appendix C), determines for itself when to stop, and then writes its output to another file on disk (attached as Appendix D).

The program is designed to run through the following phases of operation:

8.4.1 Read input from file

Up until around line 177, the software does general preparation (such as declaring variables) and also reads the input data into memory. Once having read these data, there are no further disk operations until the output is finally written to another file on disk.

8.4.2 Test ‘risk of error’ loop

Having read the data, the next principle phase of operation is for the system to measure the estimating variability inherent within the project, as defined by this data, and in accordance with the model that defines the ‘risk of being wrong’, as explained in Chapter 5. This process is depicted below.

```

for (item = 0; item < NUMBER_OF_ITEMS; item++) {
    double[] wrongResults = new double[NUMBER_OF_TESTS];
    for (i = 0; i < NUMBER_OF_TESTS; i++) {
        // 1. the contractor could be wrong as regards their estimate of the
        appropriate discounting rate...
        double simulatedDiscFactor = normrand(discFactor, discSD);

        // 2. also when estimating the variance in item's quantity...
        double simulatedExpectedVariation =
normrand(expectedVariation[item], varSD[item]);

        // 3. also when estimating each workgroup's escalation rate...
        double simulatedEscFactor =
normrand(escFactor[workGroup[item]], escSD[workGroup[item]]);

        // 4. also with their scheduled timing of the project...
        float[] simulatedLambda = simulateLambda(lambda[item],
confidentOfScheduledTiming[item]);

        for (int month = 0; month < NUMBER_OF_MONTHS; month++) {
            wrongResults[i] += Math.pow(simulatedDiscFactor,
month+1) * simulatedLambda[month] * billQuantity[item] *
simulatedExpectedVariation * Math.pow(((simulatedEscFactor - 1) *
HAYLETT_FACTOR + 1), month+1) * (1.0 - rateRetention[month]);
        }
    }
    // determine the mean and SD of the wrongResults[]
    wrongMean[item] = mean(wrongResults);
    wrongSD[item] = stdev(wrongResults);
}

```

8.4.3 Scaling risk loop

The system next embarks on a phase where it seeks to determine the extent of the risk in each item, as well as in the project as a whole. It dedicates the first 20,000 iterations of the ‘main loop’ (described below) to measure these. This exercise

serves to then facilitate that the risks can be rescaled in the manner that a risk of quantum 0 becomes the equivalent of the highest risk encountered during these 20,000 tests, whilst a risk of 1 becomes the equivalent of the least risk discovered. Any (more extreme) risk subsequently found to translate to values outside of this 0-1 range gets effectively truncated to 0 or 1, as the case may be, which serves the purpose adequately. In effect, prices outside of this range never become serious contenders for the ultimately chosen pricing of the bills of quantities anyway – and so the value of risk assigned to these does not so much matter besides providing recognition that the pricing is so extreme as not to warrant further consideration.

8.4.4 Main test loop

After having rescaled the risks (to the 0-1 scale described above) the main loop is restarted, with the potential of running two million times. Memory arrays are dimensioned to store some results from each of these 2m iterations (simply for the purposes of reporting them, with a view to being able to better appreciate the behaviour of this system). If this loop runs this full distance, it will terminate and treat the best result that it will have found by then as the quasi-optimum recommendation for the contractor. If, however, instead, the system determines for itself that it has prior to the expiration of the full duration, that further iterations seem unlikely to be able to improve on the best result that it has already found, it terminates this loop, writes the results to disk and shuts down.

In the example reported on, as a stochastic test of the hypothetical project (as described by the input data presented in Appendix B) the main loop ran around 1,150,000 number of iterations, taking 3 days before shutting down.

This main loop runs from line 241 to line 828.

This loop is run in phases called “generations” that are of 1000 iterations duration. Each generation runs according to settings that are determined at the outset of that generation, at the hand-over from the previous generation. These settings are

decided by the previous generation on the basis of the knowledge that has been accumulated to that stage.

The first ‘generation’, i.e. the first 1000 iterations, entails pricing being decided randomly because as yet there is no intelligence. These first 1000 results are analysed and serve as the catalyst for determining the settings that will govern the second generation, and so on. Within the first generation, the first three iterations are treated as special cases, designed to serve to generate data that will prove useful for the system itself to better understand the project it has begun to process.

The first iteration tests the case of assigning to each item, the lowest prices that the contractor has indicated as being possibly acceptable in the marketplace.

```
if(iteration == FIRST_TEST)
    itemPrice[item] = origLoPriceGuide[item];
```

These obviously do not summate to the tender price but this constraint is lifted for the duration of this iteration to facilitate this test. The effect of this test is to provide a measurement of the risk at the extreme edge of low prices, both for individual items as well as for the project as a whole.

The second iteration provides a test of the highest possible prices, at the opposite end of the scale.

```
else if(iteration == SECOND_TEST)
    itemPrice[item] = origHiPriceGuide[item];
```

The third iteration tests the scenario where all items are priced with the same mark-up, that is, a ‘balanced bid’.

```
else if(iteration == THIRD_TEST)
    itemPrice[item] = roundCents((float)(itemCost[item] * (1.0 + avgMarkUp)));
```

A balanced bid should represent a very low-risk pricing schedule and it is for this purpose that this test is being done: to measure this degree of minimum risk.

Please note that the risk with a balanced bid is not necessary a risk of *nil*, for two reasons, namely: (a) as explained in Chapter 5, the risk of *being wrong* is linear

with respect to the price and so ‘balanced’ prices (not being *nil*) still attract some of this risk; and (b) the risks of *reaction* and *rejection* are only *nil* within the central range around which the market has an expectation of this price. Most times, a balanced price will fall within this central range. However, if the contractor has an abnormal cost of any item, or if the market is accustomed to prices for an item that is commonly *loaded*, then interestingly, if any contractor were to submit a *balanced* price, this price might appear unusual and might attract attention. This argument is discussed further in Section 2.2.

These first three iterations then provide an initial 0-1 scaling of the risks involved that is then used for the duration of the balance of the first 20,000 iterations. During this period, the greatest and least risks are recorded and these then serve to rescale the risks to serve as a replacement of the base values of initial 0-1 scaling.

Pricing Sequence

Beyond the first generation, the items are priced according to a selection of five different pricing sequences. The sequencing of the pricing is very significant: items priced early in the sequence enjoy complete freedom by which their price is decided whereas those that are scheduled to be priced at the end of the sequence are assigned whatever price is necessary to ensure that the constraint is satisfied whereby all the item prices summate to the overall tender price.

The five sequencing techniques that have been adopted are as follows:

Sequences “A”, “B” and “C” – the sequences that the knowledgebase has discovered have produced the highest value of utility thus far. Each of these sequences is used $1/9^{\text{th}}$ of the time, i.e. $1/3^{\text{rd}}$ of $1/3^{\text{rd}}$. For this purpose, these sequences are recorded whenever any pricing combination gives rise to one of the best-yet *values of utility*. The choice between these three is made randomly and, because it is found that there is considerable common-knowledge that is shared between these three, there is found to be sufficient gained from this knowledge by having all three share having influence on $1/3^{\text{rd}}$ of the iterations tested.

```

if(beyondFirstGeneration && iteration % 3 == 0) {
    double seed = Math.random();
    int oneOf3 = seed < 0.33 ? 0 : seed > 0.66 ? 2 : 1;
    sequence = oneOf3 == 0 ? "A" : oneOf3 == 1 ? "B" : "C";
    for (item = 0; item < NUMBER_OF_ITEMS; item++)
        pricingSequence[item] = bestPricingSequences[item][oneOf3];
}

```

Sequence “D” – 1/3rd of the iterations are assigned a random sequence accomplished by way of reshuffling the last-used sequence.

```

} else if (iteration % 3 == 1) {
    sequence = "D";
    // reshuffle the sequence of priority for pricing all items
    for (item = 0; item < NUMBER_OF_ITEMS; item++) {
        randomNumber = item +
(int)Math.floor(Math.random()*(NUMBER_OF_ITEMS - item));
        int rand = pricingSequence[randomNumber];
        for (j = randomNumber; j > item; j--) {
            pricingSequence[j] = pricingSequence[j-1];
        }
        pricingSequence[item] = rand;
    }
} else {

```

Sequence “E” – another 1/3rd of the iterations are assigned a sequence giving priority to those items that have been found to produce the most utility relative to their price. A further twist on this technique entails not pricing all the items in this manner but instead pricing some in this manner and the remainder randomly. The switch-over point between these two techniques is again chosen randomly.

```

} else {
    // price an initial portion of the items in accordance with orderItemResults
    and then randomly scramble the rest
    int extent = (int)((NUMBER_OF_ITEMS-1) * Math.random());
    sequence = "E" + extent;
    for (item = 0; item < NUMBER_OF_ITEMS; item++)
        pricingSequence[item] = orderItemResults[item%2 == 0 ? item / 2
: (NUMBER_OF_ITEMS-1) - (item / 2)]; // rank them starting from the one
that produces the greatest utility / price
    for (item = extent; item < NUMBER_OF_ITEMS; item++) {
        randomNumber = item +
(int)Math.floor(Math.random()*(NUMBER_OF_ITEMS - item));

```

```

        int rand = pricingSequence[randomNumber];
        for(j = randomNumber; j > item; j--) {
            pricingSequence[j] = pricingSequence[j-1];
        }
        pricingSequence[item] = rand;
    }
}

```

The results of the test show that all of the above techniques succeed in contributing some improvements in the overall utility that is accomplished. This is shown in Appendix D as per the following extract:

Improvement (**due to item pricing priority sequence A**) to 17073.15187770346 after 1182 efforts, increasing the reward to 6015219 (a 139% improvement on the profit from a balanced bid) whilst reducing the total risk to 84763 (riskReaction = 5820, riskOfBeingWrong = 28579 & riskRejection = 50363)

These various sequencing techniques are therefore all significant, not solely for the purpose of being successful ultimately when finding the best-producing pricing schedule, but also with soliciting the vast variety of scenarios that all contribute to the enlargement of the knowledge-base that ultimately serves as the asset necessary for the final discovery of the best scenario.

Tender Price Constraint

The following block of code serves as preparation for ensuring that in all scenarios, all the items will be priced in a manner that the constraint is satisfied that they all summate to the tender price.

```

float mult = 0.0F;
do {
    for (int ii = NUMBER_OF_ITEMS-1; ii >= 0; ii--) {
        assignedMinPrice[pricingSequence[ii]] = (beyondFirstGeneration ?
(bestLowItemPrices[kk][pricingSequence[ii]][0] - mult *
bestLowItemPriceSD[kk][pricingSequence[ii]][0]) :
origLoPriceGuide[pricingSequence[ii]]);
        assignedMaxPrice[pricingSequence[ii]] = (beyondFirstGeneration ?
(bestItemPrices[kk][pricingSequence[ii]][0] + mult *
bestItemPriceSD[kk][pricingSequence[ii]][0]) : origHiPriceGuide[pricingSequence[ii]]);
    }
}

```

```

        minimumPricingCommitment[ii] = (ii < NUMBER_OF_ITEMS-1 ?
minimumPricingCommitment[ii+1] : 0) + assignedMinPrice[pricingSequence[ii]] *
billQuantity[pricingSequence[ii]];
        maximumPricingCommitment[ii] = (ii < NUMBER_OF_ITEMS-1 ?
maximumPricingCommitment[ii+1] : 0) + assignedMaxPrice[pricingSequence[ii]] *
billQuantity[pricingSequence[ii]];
    }
    if(mult == 0)
        mult = 0.1F;
    else
        mult *= 1.1;
} while(beyondFirstGeneration && (minimumPricingCommitment[0] > tenderPrice ||
maximumPricingCommitment[0] < tenderPrice));

```

In effect, this technique uses the pricing sequence that the previous block of code will have chosen and it calculates the effects of pricing all the items, in this sequence, assigning them the lowest prices that the contractor has indicated (as input), as well as the highest prices – recording both of these scenarios separately. It does this starting from the items at the end of the sequence, working forward to the first item, and it stores these two sets of cumulative results as it progresses through the sequence.

This knowledge will serve to moderate the pricing midway through the pricing sequence, as soon as the effect of those prices already assigned to the initial items threatens to violate the tender price constraint. At some point through each sequence, the pricing will inevitably be such that, in summation, if left to proceed without moderation, they will either be higher or lower than the tender price. If it is found that the pricing is heading to be higher than the tender price, the system intercepts this scenario and, at the necessary moment of intervention, breaks from the further pursuance of the pricing and instead switches over to a mode in which all the remaining items in the sequence are assigned their minimum prices (or maximum prices). This then ensures that the tender price constraint is satisfied and this illustrates the significance of the pricing sequencing techniques described above.

Item Pricing

The system's next function is to price each item for that iterative test (of which there are two million conducted). The system begins to loop through all the items in the project,

looking to price them in the sequence that has already been chosen for this iterative test. It first checks whether it is necessary to price all of the remaining items in the sequence with their minimum or maximum prices.

```

if(priceAllRemainingItemsWithMinimumPrice) {
    itemPrice[item] = roundCents(beyondFirstGeneration ? assignedMinPrice[item] :
origLoPriceGuide[item]);
    itemPriceSD[item] = 0;
} else if(priceAllRemainingItemsWithMaximumPrice) {
    itemPrice[item] = roundCents(beyondFirstGeneration ? assignedMaxPrice[item] :
origHiPriceGuide[item]);
    itemPriceSD[item] = 0;
} else {

```

Next, the system determines whether the item in question is one that produces a relatively high level of utility (per unit price) or a relatively low level of utility. It has been decided to treat those that fall into the top third of items as the better ‘producers’ and those that fall into the bottom third as those that are the worse producers.

```

boolean bestProducersOfUtility=itemRanking[item]<(float)NUMBER_OF_ITEMS/3F;
boolean worstProducersOfUtility =
itemRanking[item]>(float)NUMBER_OF_ITEMS*2F/3F;

```

This knowledge is used in the next step which entails deciding on whether to experiment by ‘loading’ or ‘unloading’ the price of the item in question. The technique used for this entails randomly switching between three different methods by which to make this decision.

```

if(pickMethod(itemBeta[item] > (mean(itemBeta) + 0.5 * stdev(itemBeta)),
mean(pricingSchedule[item])>(origHiPriceGuide[item] + origLoPriceGuide[item])/2.0F,
bestProducersOfUtility)) { // load these
    loadItem = true;
    direction = +1;
} else if(pickMethod2(itemBeta[item] < mean(itemBeta), mean(pricingSchedule[item]) <
(origHiPriceGuide[item] + origLoPriceGuide[item]) / 2.0F,
worstProducersOfUtility)) { // unload these (i.e. drain these to provide funding for the
loading)
    loadItem = true;
    direction = -1;
}

```

As can be seen from the above code, the first of these methods entails identifying whether the item has an *itemBeta* value of more than $mean(itemBeta) + 0.5 * stdev(itemBeta)$ where $mean(itemBeta)$ is the calculated mean of the *itemBeta* values for all the items, and $stdev(itemBeta)$ is the calculated standard deviation. The second method entails identifying the items that the system has discovered are best being allocated prices higher than the price halfway between the low and high pricing extremes that the contractor has provided as input for this item. It makes this assessment based on the knowledge accumulated in the *pricingSchedule[]* array, which contains the prices of the best combinations of prices found up until that stage. Notice that this method assumes that in the event that one of the best producing schedules has done well on the basis of this item having been allocated a relatively high price, then this is most-likely indicative of the fact that this item makes good use of a high-price allocation, relative to other items. The third method makes use of the assessment of the *bestProducersOfUtility* and the *worstProducersOfUtility*, as discussed above. By randomly switching between these methods facilitates, in the style of Fuzzy Logic, that the system need not be definitively and consistently correct as regards any one of these assessments. If the system is ‘wrong’ in this regard, it will only result in a waste of time processing iterations that have no chance of contributing to any improvement in the knowledgebase, let alone provide prospect of the best combination of prices. This potential wastage has been monitored, partially by way of analysing the feedback from the system (as shown in Appendix D) and also by running this software in a ‘debugging’ mode in which each step of the process can be thoroughly viewed and monitored. It has been found that, collectively, all of these methods serve well to guide the process towards the accumulation of the necessary knowledge as well as towards the ultimate optimum.

Consider the tender price to be a ‘finite resource’ that can be distributed between a project’s items in compliance with a set of constraints. When testing alternative combinations of prices, whenever any one item is assigned a test price, this assignment imposes a further constraint to some degree on what prices can be assigned to the remaining items. All of the item prices need to summate to the tender price, so if the pricing assigned to the items early on in the sequence are high prices, this will force a need for low prices later on the sequence so as not to violate the tender price constraint.

In this sense, it is useful to think of the tender price as a *resource* that has a fixed value and that must be spread over all of the items that are the constituent components of the overall project.

Essentially, all of these methods strive to ensure that higher prices will be assigned to those items that will make best use of this (finite resource), whilst funding this assignment by way of stripping down (or ‘unloading’) the prices of those items where the pricing is of least significance (i.e. those with the least ability to contribute a relatively competitive return for that amount of use of the overall pricing allocation).

By the end of this step, the system has therefore split all of the items into three groups, namely:

- those that the system will test (for the duration of that one iteration – of which there are ultimately two million conducted) with a high price,
- those that instead will be tested with a low price, and
- those that fall between these two groups, where the system is less certain.

This split is then reassessed with each iteration.

Loading of prices

In the next step of the process, the test prices for each item are decided. These are chosen stochastically given the mean and standard deviation of the range chosen to explore. This mean and standard deviation for each item are assigned the variable names *guidePrice* and *guideSD* in the following extracted block of source code. This extract is specific to the situation where the item has been chosen to receive a (positively) loaded price. A very similar process applies to those items that have been chosen to receive (unloaded) low prices.

```
guidePrice = pickrand(mean(bestItemPrices[kk][item]) + direction * alpha
*bestItemPriceSD[kk][item][0],
    bestItemPrices[kk][item][0] + direction * alpha * bestItemPriceSD[kk][item][0],
    bestItemPrices[kk][associatedItem(item)][0],
```

```

bestItemPrices[kk][r][0],
pricingSchedule[item][0] + direction * alpha * pricingScheduleSD[item][0],
mean(pricingSchedule[item]) + direction * alpha * pricingScheduleSD[item][0],
origLoPriceGuide[item] + (2.0+direction * alpha)/4.0 * (origHiPriceGuide[item] -
origLoPriceGuide[item]),
mean(bestItemPrices[kk][item]),
mean(pricingSchedule[item]),
(origHiPriceGuide[item] + origLoPriceGuide[item]) / 2.0F);
guideSD = (float)(beta * pick(bestItemPriceSD[kk][item][0] * 3,
bestItemPriceSD[kk][item][0],
bestItemPriceSD[kk][associatedItem(item)][0],
bestItemPriceSD[kk][r][0],
pricingScheduleSD[item][0],
pricingScheduleSD[item][0],
(origHiPriceGuide[item] - origLoPriceGuide[item]) / 3,
bestItemPriceSD[kk][item][0],
pricingScheduleSD[item][0],
(origHiPriceGuide[item] - origLoPriceGuide[item]) / 2.0F));

```

This code shows that the *guidePrice* and *guideSD* are chosen from a random selection of ten different alternatives, namely:

- a) $mean(bestItemPrices[kk][item]) + direction * alpha * bestItemPriceSD[kk][item][0]$
and $beta * bestItemPriceSD[kk][item][0] * 3$, respectively

where *direction* is +1 for loaded items, and -1 for unloaded items

alpha and *beta* are settings decided by the system for each *generation* of tests (see below)

$mean(bestItemPrices[kk][item])$ is the calculated mean of the 10 prices found, up until that stage in the process of the testing, to be those that produced the highest value of utility for that item

$bestItemPriceSD[kk][item][0]$ is the standard deviation used in the stochastic selection the first time that this price was identified.

- b) $bestItemPrices[kk][item][0] + direction * alpha * bestItemPriceSD[kk][item][0]$
and $beta * bestItemPriceSD[kk][item][0]$, respectively

where $bestItemPrices[kk][item][0]$ is the price that has been found, up until that stage in the process of the testing, to be the one that produced the highest value of utility for that item

- c) $bestItemPrices[kk][associatedItem(item)][0]$
and $beta * bestItemPriceSD[kk][associatedItem(item)][0]$, respectively

where $bestItemPrices[kk][associatedItem(item)][0]$ is the best price (thus far) for a randomly chosen *associated item* of the item in question, where *associated items* are items found to attract very similar (within R1.00 of each other) prices. In other words, it has been found (up until that stage in the process) that the best prices suited to these items are very similar. For this reason, the system links such items as being possibly ‘related’ to each other (i.e. sharing similar characteristics) with the benefit that any intelligence gained on one such item may be of value if shared with the others. The identification of *associated items* is explained below.

- d) $bestItemPrices[kk][r][0]$
and $beta * bestItemPriceSD[kk][r][0]$, respectively

where r is a random choice of another item: indicating a wild-shot that two items may be discovered, purely by chance, to share enough in character that the knowledge gained on the one item has value being shared with the other

- e) $pricingSchedule[item][0] + direction * alpha * bestItemPriceSD[kk][item][0]$
and $beta * pricingScheduleSD[item][0]$, respectively

where $pricingSchedule[item][0]$ is the price that was assigned to this item at the time that the system discovered the most utility from any one combination of prices

$pricingScheduleSD[item][0]$ was the standard deviation used in the process of discovering this price the first time that it was discovered

- f) $mean(pricingSchedule[item]) + direction * alpha * pricingScheduleSD[item][0]$
and $beta * pricingScheduleSD[item][0]$, respectively

where $mean(pricingSchedule[item][0])$ is the calculated mean of the prices that were assigned to this item at the time that the system discovered the most utility from the best three combinations of prices

- g) $origLoPriceGuide[item] + (2.0 + direction * alpha) / 4.0 * (origHiPriceGuide[item] - origLoPriceGuide[item])$ and $(origHiPriceGuide[item] - origLoPriceGuide[item]) / 3$, respectively

where $origLoPriceGuide[item]$ is the price that the contractor provided as input as being the lowest price that they would consider likely to be acceptable

$origHiPriceGuide[item]$ is, similarly, the highest price that the contractor consider likely to be acceptable

- h) $mean(bestItemPrices[kk][item])$ and $bestItemPriceSD[kk][item][0]$, respectively

- i) $mean(pricingSchedule[item])$ and $pricingScheduleSD[item][0]$, respectively

- j) $(origHiPriceGuide[item] + origLoPriceGuide[item]) / 2.0F$ and
 $(origHiPriceGuide[item] - origLoPriceGuide[item]) / 2.0F$, respectively

The following chart (Figure 8.1) shows the effectiveness of each of these techniques in the test that has been conducted, as regards the number of times that they have identified prices that were amongst the ten best for that item, as at that stage. This shows that all of these techniques all manage to contribute to the knowledgebase and are, hence, all valuable albeit that some of the techniques are, in the instance of this test, more productive in this regard than some others. The techniques identified as 'j' and 'd' are found to be the most productive, but these are relative 'wild shot'-styles of technique compared to the others. Knowledge gained from this nature of technique is then, in effect, 'spread' across the knowledgebase, and shared with other associated items, when appropriate, by way of techniques such as those identified as 'c'. In this manner, these different techniques can be viewed as being complementary with each other - aiding each other to build the knowledgebase.

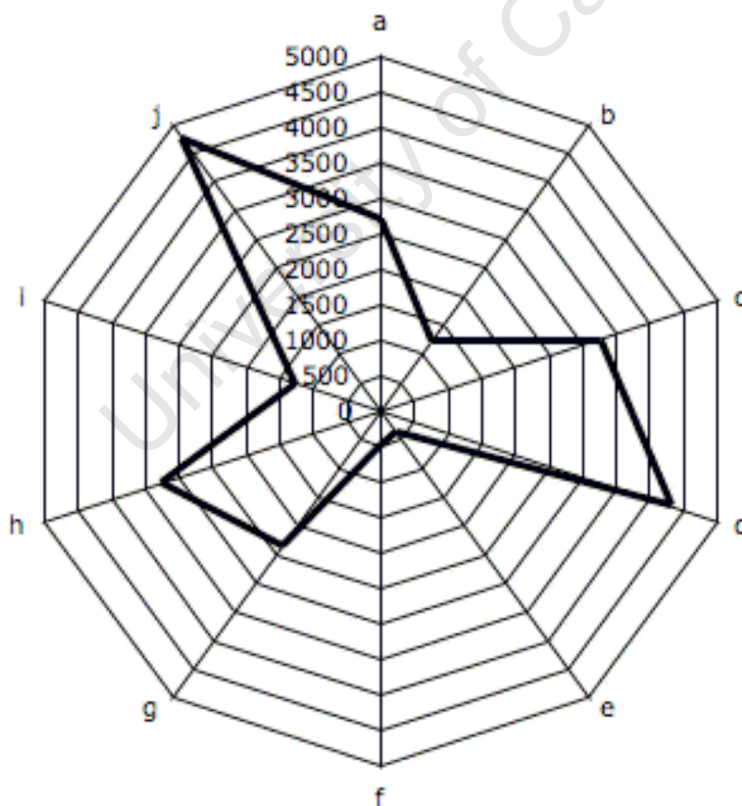


Figure 8.3 The relative effectiveness of the different methods of identifying the best item prices, in the case of the test of the hypothetical project

The items that fall into the set that lies between the ‘loaded’ and ‘unloaded’ ones, are similarly assigned prices according to the following extracted block of code:

```

guidePrice = pickrand(bestItemPrices[kk][item][0],
    bestItemPrices[kk][associatedItem(item)][0],
    bestLowItemPrices[kk][associatedLowItem(item)][0],
    pricingSchedule[item][0],
    mean(pricingSchedule[item]),
    mean(bestItemPrices[kk][item]),
    (origHiPriceGuide[item] + origLoPriceGuide[item]) / 2.0F,
    (origHiPriceGuide[item] + origLoPriceGuide[item]) / 2.0F);
guideSD = (float)(beta * pick(bestItemPriceSD[kk][item][0],
    bestItemPriceSD[kk][associatedItem(item)][0],
    bestLowItemPriceSD[kk][associatedLowItem(item)][0],
    pricingScheduleSD[item][0],
    pricingScheduleSD[item][0],
    bestItemPriceSD[kk][item][0],
    1.2F * (origHiPriceGuide[item] - origLoPriceGuide[item]) / 2.0F,
    (origHiPriceGuide[item] - origLoPriceGuide[item]) / 2.0F));

```

These items are, therefore, in effect given a random selection of eight different possible prices. Keep in mind that in all of these instances, the objective is to ‘drain’ these items, assigning them low prices, at the bottom-end of their scales, so as to, in effect, ‘fund’ the ability by which higher prices can be assigned to other prices. The array named *bestLowItemPrices* refers to the prices at the bottom-end of the scale, found to be best suited to items, in the event that this item is serving this role of being ‘unloaded’ in the interests of others being ‘loaded’.

The results (discussed in the next chapter) indicate that all of the above methods contribute to the solicitation of useful intelligence.

Cross-pollination of knowledge between ‘associated’ items

The system’s artificial intelligence is designed to hypothesise that there is the prospect of some items being similar to other items. For instance, two billed items may be so similar that the contractor effectively views them as being virtually identical to each other, albeit that the Standard System of Measurement has given cause for the client’s quantity surveyor to separate them. In practice, items of formwork to soffits of slabs, and sides and soffits of beams typically fall into this category. The system could be improved upon

by allowing the contractor to give input as to indication of these associated items, but the software developed here looks to identify these situations without any such explicit input.

It is supposed that in such an instance, the contractor will have good reason to treat the pricing of such items similarly, if not identically the same. On this basis, the system has been designed to share knowledge gained on any one such item, with all other items that may be associated with it. If the system discovers that one such item is performing best when assigned a price of (say) R100.00, it is presumed that the system will benefit from this intelligence being shared that all other similar prices be tested with the same, or similar, prices.

The system is designed to search for items that may be similar and to do so whenever any one *generation* hands over to the next one. It looks for items that are becoming drawn to prices that are within R1.00 of each other. It supposes that if two similar items share enough as to be performing best at (say) R50.00 and R50.80, respectively, that it will henceforth be worthwhile considering the merits of pooling the knowledge on these two items, such that if the one ultimately, gravitates to a price of (say) R50.32 as being its best-performing price, the other item might benefit from sharing this knowledge, and vice versa.

The following block of code serves to link items as being “associated” with each other:

```

if(ii == 0 && birthOfNextGeneration) {
    for (int ai = 0; ai < NUMBER_OF_ITEMS; ai++) {
        for (int aj = 0; aj < NUMBER_OF_ASSOCIATES; aj++) {
            assItem[ai][aj] = Integer.MAX_VALUE;
            assLowItem[ai][aj] = Integer.MAX_VALUE;
        }
        int associate = 0;
        int lowAssociate = 0;
        for (int aj = 0; aj < NUMBER_OF_ITEMS; aj++) {
            if (((bestItemPrices[kk][ai][0] >= bestItemPrices[kk][aj][0] &&
bestItemPrices[kk][ai][0] < bestItemPrices[kk][aj][0]+1) || (bestItemPrices[kk][ai][0] <=
bestItemPrices[kk][aj][0] && bestItemPrices[kk][ai][0] > bestItemPrices[kk][aj][0]-1))
&& associate < NUMBER_OF_ASSOCIATES && ai != aj) { // priced within a $ of each
other
                assItem[ai][associate] = aj;
                associate++;
            }
        }
    }
}

```

```

    }
    if (((bestLowItemPrices[kk][ai][0] >=
bestLowItemPrices[kk][aj][0] && bestLowItemPrices[kk][ai][0] <
bestLowItemPrices[k][aj][0]+1) || (bestLowItemPrices[kk][ai][0] <=
bestLowItemPrices[kk][aj][0] && bestLowItemPrices[kk][ai][0] >
bestLowItemPrices[k][aj][0]-1)) && lowAssociate < NUMBER_OF_ASSOCIATES &&
ai != aj) { // priced within a $ of each other
        assLowItem[ai][lowAssociate] = aj;
        lowAssociate++;
    }
}
}
}

```

Setting things up for the next generation

As each *generation* hands over to the next one, the system also looks to reassess its settings, with the goal of making the next generation as productive of new intelligence as is possible. Two settings have been chosen to govern this evolutionary process: called *alpha* and *beta*. *alpha* determines the extent to which the next generation will stray away from the range of prices being tested previously, and *beta* serves to determine how narrowly focussed, or otherwise, this range of prices will be.

alpha and *beta* serve as multiples when determining each test price in the stochastic manner described above. Higher values of *alpha* serves to shift the ‘range of exploration’ into new territory, in the hope of new discoveries, whilst lower values serve to ensure that the system more-so repeatedly tests the use of prices very similar to those that have been tried before – serving to both refine the choice of the best-price (getting to the stage of choosing the exact cent value that will serve best) as well as help find the best combination of these prices. This choice as regards the best combination is also one that entails the decision as regards the sequence of the pricing process – or, in other words, the decision as regards which items to price as priorities in advance of others, choosing to treat some items’ pricing as being less important than others. Those that are found to be more important are priced early in the sequence, giving them an unrestrained freedom, whilst those at the end of the sequence have their freedom removed and their price dictated by the need to satisfy the tender price constraint.

The code that decides the *alpha* and *beta* values for the next generation, is as follows:

```
String route = "a";
if(recentImprovement || ((beta >= (stillLearning[0] / eps) && alpha >= (stillLearning[1] /
eps)) && (beta < (stillLearning[0]+0.2)*eps || alpha < (stillLearning[1]+0.2)*eps))) {
    route += "b";
    if(recentImprovement) {
        route += "c";
        lastUsefulBeta = beta;
        lastUsefulAlpha = alpha;
        if(alpha > 0)
            alpha -= 0.1F;
    } else {
        route += "d";
        if(beta >= (stillLearning[0]+0.2)/eps && alpha >=
(stillLearning[1]+0.2)/eps) {
            route += "e";
            if(beta > alpha)
                beta = 0;
            else
                alpha = 0;
        } else if(beta >= (stillLearning[0]+0.2)/eps) {
            beta = 0;
            if(alpha >= (stillLearning[1]+0.2)/eps)
                alpha = 0;
            else
                alpha += 0.1F;
            route += "f";
        } else if(alpha >= (stillLearning[1]+0.2)/eps) {
            alpha = 0;
            beta += 0.1;
            route += "g";
        } else {
            beta += 0.1;
            alpha += 0.1F;
            route += "h";
        }
    }
} else if(notSoRecentImprovement) {
    route += "i";
    beta = 0;
    alpha = 0;
    notSoRecentImprovement = false; // reset
} else { // if we seem to be going nowhere, we now need to embark on a new regimen by
use to make good use of the intelligence that has already been gathered
    route += "j";
```

```

    if(similar(beta, lastUsefulBeta) && similar(alpha, lastUsefulAlpha)) {
        route += "k";
        beta = 0.1F;
        alpha = 0.0F;
    } else if (Math.random() < 0.5) {
        route += "l";
        beta = 0;
        alpha = 0;
    } else {
        route += "l";
        beta = lastUsefulBeta;
        alpha = lastUsefulAlpha;
    }
}
beta = roundOff(beta);
alpha = roundOff(alpha);
if(pickDebug.length() > 0) {
    System.out.println();
    System.out.println(pickDebug); pickDebug = "";
    if(bestRecentResult != bestResult)
        System.out.println("Best recent result = "+bestRecentResult);
    System.out.println(iteration+": Changed beta to "+beta+" & alpha to "+alpha+"
(having taken route "+route+)");
    recentImprovement = false; // reset every generation, regardless
    bestRecentResult = 0;
    numBubbles = 0;
}

```

In effect, this code determines that the system gains the following characteristics in its pursuit of greater intelligence:

- If the system has recently discovered a pricing combination that produces the best-yet value of utility for the contractor, it retreats and elects to continue to explore the territory defined by way of the same settings, even reducing the value of *alpha* by 0.1, knowing that the logical progression in future *generations* will involve incrementing *alpha* again, moving back over this same territory a second time,
- If, however, the system has not found any new overall improvement in the project's utility, it checks to see how recently *alpha* and *beta* have led to any new knowledge. If neither has been successful lately at generating any fresh

knowledge that has proven itself to be useful, it decides to change one of them radically, so as to turn the exploration in a new direction,

- If the system has not found any new knowledge of use lately, but had done previously, in the not too distant past, it resets *alpha* and *beta* both to 0, so as to focus in on trying to marginally squeeze out slightly more from the knowledge already gained,
- If no new knowledge has been learnt for a 'long' time (as judged by the system itself), it randomly switches between further exploring the best knowledge already gained (using the *alpha* and *beta* values of 0) and going back to reusing the settings of *alpha* and *beta* that were in place at the time at which the last new useful knowledge was gained, and
- Ultimately the system shuts down if it detects that there is little prospect of any further improvement, in which case it then declares that the best result that it has found must constitute a reasonable estimate of the optimum pricing combination; it records its results to disk and closes down.

Assessing the pricing

The effectiveness of the pricing is assessed on the basis that the *value of utility* (unique to this contractor) that this pricing generates. This utility is determined on the basis of the calculated profitability and risk for each item, as well as for the combination of items, for each iterative test, of which two million are conducted and recorded.

8.4.5 Write output to file

The final phase of operation for the system is to write its results to a data file on disk, and then shut down. In the interim, as it proceeds, it reports on its progress in real-time through the terminal interface. An example of the former is attached as Appendix D and an example of the latter is attached as Appendix C.

The output has been further dissected by way of a utility that has been written to strip down the two million results to a sample small enough that it can be read into a

spreadsheet. The test results shown in the next chapter are produced by way of that spreadsheet analysis.

8.5 SUMMARY

This chapter has described the software that the author has written for the purposes of implementing CUP theory and testing it on a hypothetical project. The full source code of this software, written in Java, is included in Appendix A whilst the data describing the hypothetical project is included in Appendix B. The next chapter describes the test that has been conducted.

The software implements AI, GP and Fuzzy Logic in a manner as to guide a process of MCS. The manner in which the software is constructed is not suited to an industrial, practical application but it does well to serve the purposes of this academic research. It is, for instance, far slower than would be required for everyday use. Nevertheless, the objective of this work is not to develop a commercial system and what is described here now provides a basis from which a commercial derivative may evolve.

REFERENCES

- Banzhaf, W., Nordin, P., Keller, R.E., and Francone, F.D.** (1998) *Genetic programming : an introduction : on the automatic evolution of computer programs and its applications*. 2nd ed., San Francisco: Morgan Kaufmann.
- Crevier, D.** (1993) *AI: The Tumultuous Search for Artificial Intelligence*, New York, USA: BasicBooks.
- Davis, L.D. and Mitchell, M.** (1991) *Handbook of genetic algorithms*. New York: Van Nostrand Reinhold.
- Glasserman, P.** (2003) *Monte Carlo methods in financial engineering*. Berlin: Springer 2003, ISBN 0-387-00451-3.
- Hertz, D.B.** (1964) Risk analysis in capital investment. *Harvard Business Review*, no.42, pp. 95-106. Available at <http://harvardbusinessonline.hbsp.harvard.edu/b02/en/common/emailfriend.jhtml;jsessionid=VFWWCBHNEV1KAAKRGWDR5VQBKE0YIISW?productID=79504> [accessed 2nd June, 2009].
- Holland, J.H.** (1992) Genetic algorithms. *Scientific American*. July. pp.66-72.
- Metropolis, N.** (1987) The beginning of the Monte Carlo method. *Los Alamos Science*, Special issue, pp. 125-130. Available at <http://library.lanl.gov/cgi-bin/getfile?00326866.pdf> [accessed 2nd June, 2009].
- Moore, G.E.** (1965) Cramming more components into integrated circuits. *Electronics*, vol.38, no.8, April 19. Available at ftp://download.intel.com/museum/Moores_Law/Articles-Press_Releases/Gordon_Moore_1965_Article.pdf [accessed 12th June, 2009].
- Novak, V., Perfilieva, I. and Mockor, J.** (1999) *Mathematical principles of fuzzy logic*. Boston: Kluwer.
- Rubinstein, R.Y. and Kroese, D.P.** (2007) *Simulation and the Monte Carlo method*. 2nd Ed. New York: Wiley. ISBN: 978-0-470-17794-5

Watson, M. (2002) Practical Artificial Intelligence Programming in Java, Available at [http://www.kbb.waw.pl/epg/.ebooks/\(ebook\).Practical.AI.in.Java.pdf](http://www.kbb.waw.pl/epg/.ebooks/(ebook).Practical.AI.in.Java.pdf) [last accessed 9th July 2009].

Zadeh, L. (1965) Fuzzy sets. *Information and Control*, vol. 8. pp. 338-353.

University of Cape Town

9. TESTING THE MODEL

This chapter describes the test done using the software that has been developed, and described in Chapter 8. The program is tested using the data from the hypothetical project, as described in Appendix B.

The test provides feedback in real-time, over the course of the week that it takes to run, and finally it presents its results by writing them to disk.

As the system progresses, the following feedback is presented so that it can be monitored:

- the iteration index (keeping in mind that it might choose to run for the full two million iterations) – see ‘A’ below (in Figure 9.1) in the sample presented,
- the current settings of *alpha* and *beta*, being applied to the current *generation* of tests – see ‘B’ below,
- any improvements in the overall level of *utility* that is being accomplished (together with the corresponding values of return and risk) – see ‘C’,
- the method of sequencing used to produce any such improvement – see ‘D’,
- any occasion when an item’s price has produced utility for that item that is amongst the ten best-yet records of utility for that item (reported on as 0-9 where 0 is indicative of the best-yet result) – see ‘E’,
- an index reference to the pricing method used to produce this good item result – see ‘F’,
- an indication of whether this result is a function of the price being ‘loaded’ (+), ‘unloaded’ (-) or left to be priced amongst the remaining items (‘ ’) – see ‘G’.

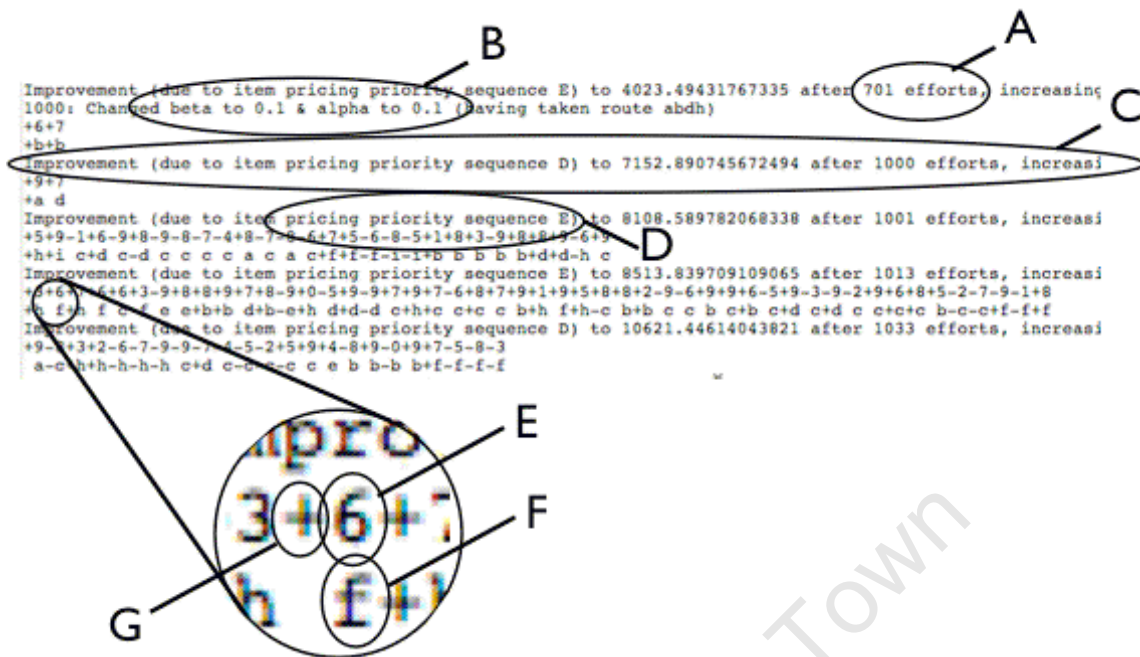


Figure 9.1 Sample of output, of which more is shown in Appendix D

A sample of this feedback is provided as Appendix D. The total volume of data generated in this manner is not provided seeing as it would equate to several hundred pages if printed.

Viewing this feedback, as the system progresses, provides a fascinating insight into the ‘psychology’ of the system – how and why it takes different tacks in its endeavour to acquire useful intelligence that should ultimately serve to identify the quasi-optimum best price set.

Notice, for instance, from the above tiny snapshot how the utility improves when the system switches over to the new generation after 1000 iterations (‘efforts’). This is indicative of the system’s largely random pricing method that it employs for the duration of the first *generation*, until this *generation* has served to provide the first insight into the project and how it might best be priced. New intelligence is then adopted as from iteration 1000, reflecting in the improved performance.

The early stages are characterised by big steps in improvement in the pricing, and the improvements become progressively smaller, and further apart, as it finds it becomes more and more difficult to incrementally continually improve.

The psychology of the system is such that it will pursue improved knowledge at the expense of a realistic chance of finding the best pricing combination. It does this as an investment in its 'intelligence' asset, seeing as without this, the system will be unlikely to discover nearly as good an ultimate best pricing combination than had it discarded the value in 'wandering off' to pursue 'roads less travelled'. For instance, very high values of *beta* and *alpha* will, almost certainly, not produce any short-term improvement in the overall pricing. And yet, the system will decide to conduct many of its tests whilst using high *alpha* and / or *beta* values, knowing that these are very effective at making new discoveries that lead to expansions of the knowledgebase.

The expansions to the knowledgebase can be monitored, to some degree, by way of watching the values being produced that are referenced as 'E', 'F', and 'G' above. The operator of the system can witness the productivity from each generation as regards the discoveries of improvements in the pricing of individual items.

Figure 9.2 shows typical progress of the system in the early stages of discovery, as measured by way of the overall utility generated from the various pricing combinations tested. Notice that the testing during the first *generation* of 1000 iterations entails pricing chosen randomly, and thereafter, the pricing becomes influenced by way of the system's interpretation of the knowledge gained thus-far.

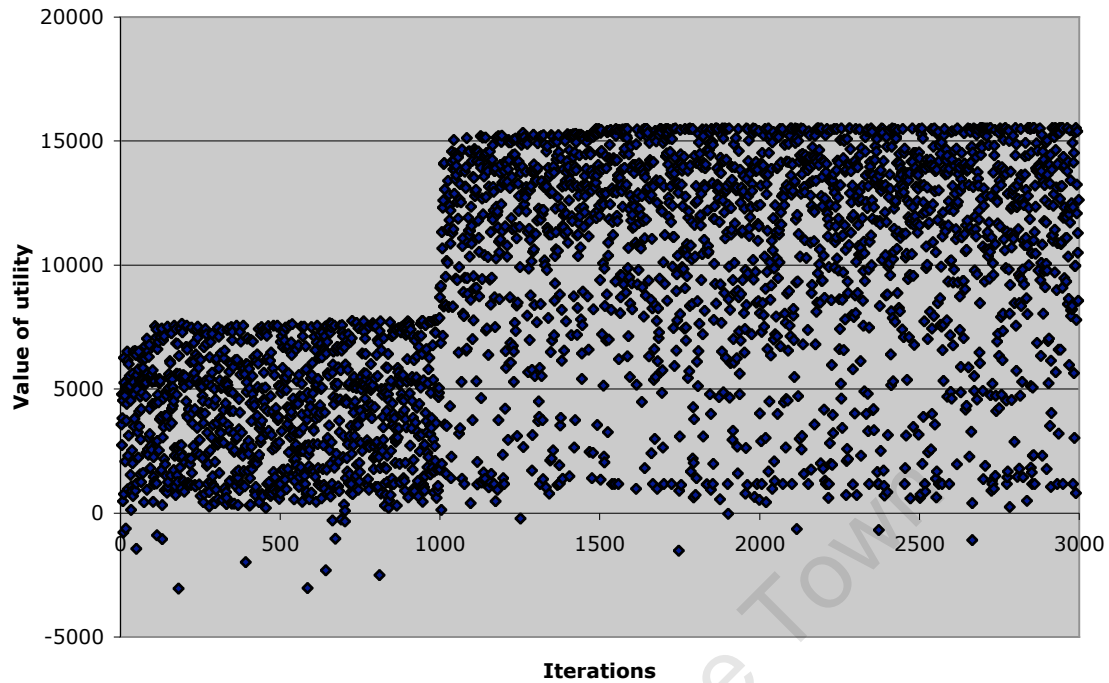


Figure 9.2 The utility produced from the first 3000 iterations tested

Figure 9.3 below shows the how the system progresses as regards increasing the return from the project, whilst seeking to bring down the risk.

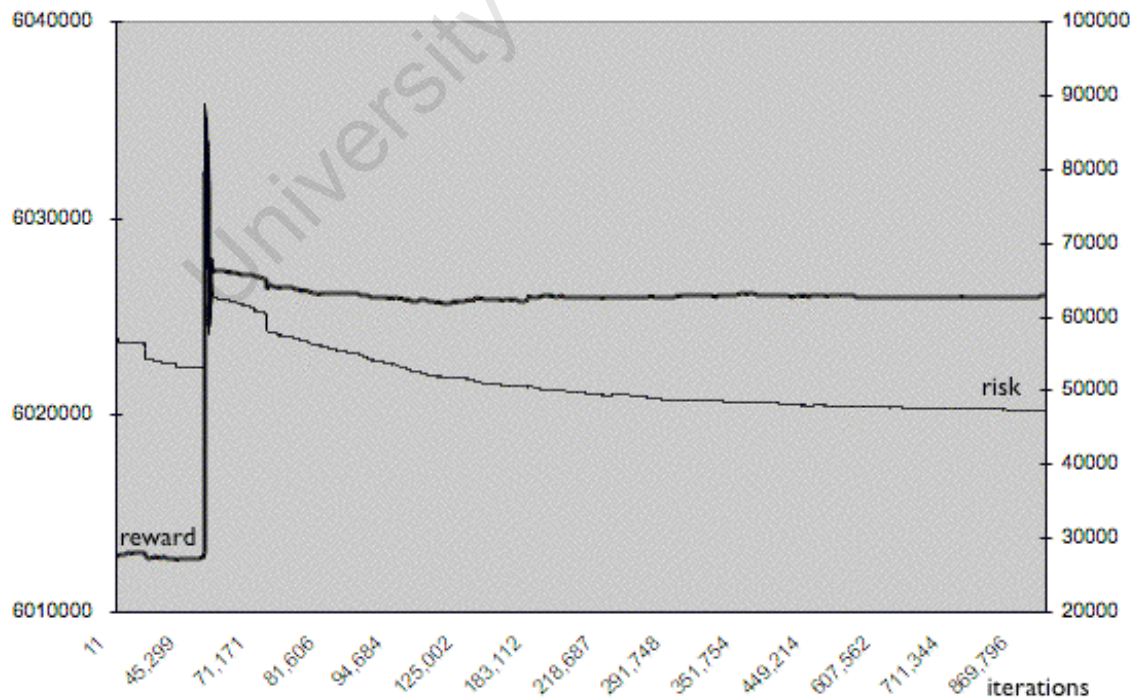


Figure 9.3 The progress in increasing rewards whilst lessening the risk

Figure 9.4 shows the progress of the system in terms of the amount of new intelligence gathered as measured by way of the number of individual item prices found to be the best performing in the course of the prior *generation* (i.e. 1000 iterations). This (stacked) chart also shows the quality of this knowledge in terms of recognising that a price that generates the highest utility thus far ('0') is better than one that generates (say) the second-best utility this far ('1'). This chart is focussed in on the early performance of the system.

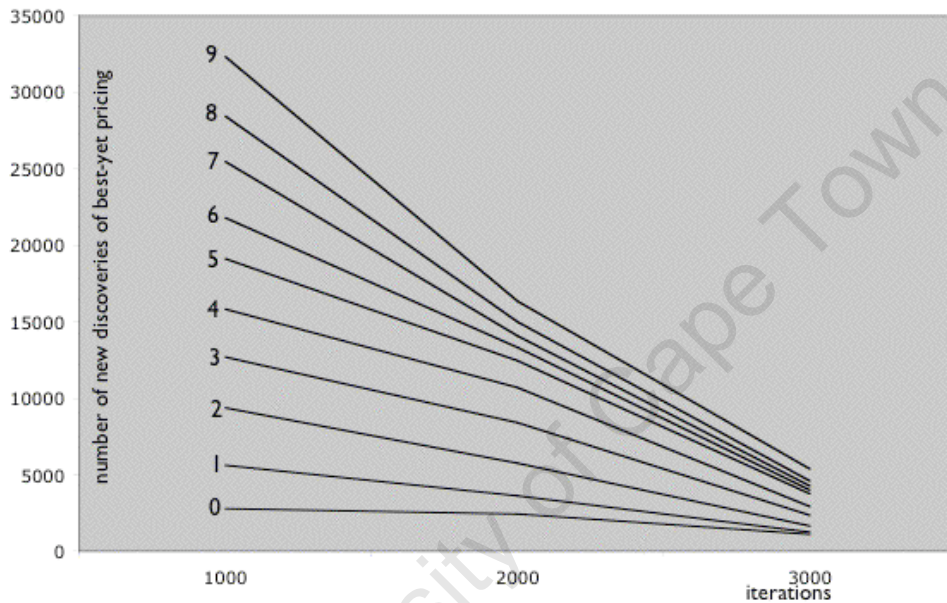


Figure 9.4 The early-stages of progress towards finding the best-yet prices for each item

Figure 9.5 shows the effects of the *alpha* and *beta* settings over approximately 60,000 iterations.

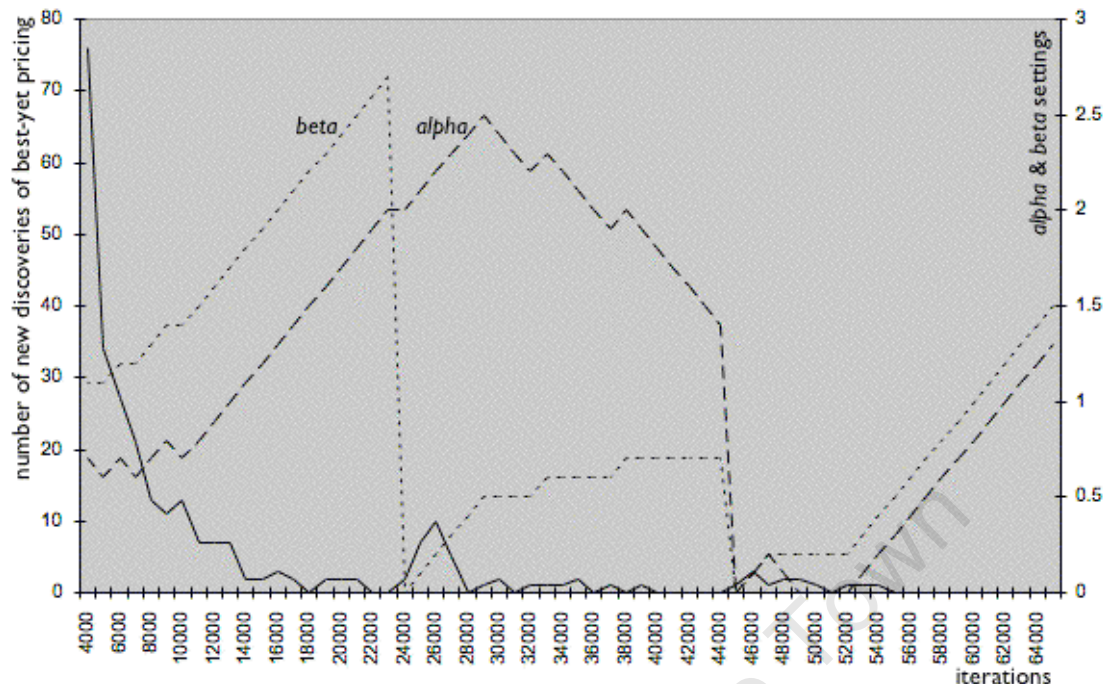


Figure 9.5 The effect of *alpha* and *beta* settings (that the system chooses for itself) on the discovery of best-yet pricing

Next, consider the individual item prices that the system has identified, as being best suited to this project. Notice that, with reference to the characteristics assigned to each of these items, as outlined in Appendix B, the system has priced each of them appropriately. No pricing falls outside of the range that the contractor indicated, as their input, as being reasonable. The full list of item prices that the system has identified is provided as Appendix E.

Ultimately, the system has succeeded in identifying a combination of item prices that is expected to generate a profit with an expected mean of around 150% more than that which would be accomplished by way of a balanced bid. The contractor stands to make a present-day profit that has a mean of R187,391 rather than R76,109, as shown in Figure 9.6 below (in both instances taking account of an assumption that the profits will be eroded by way of a “winner’s curse” by which the eventual actual costs will be 3% more than those estimated).

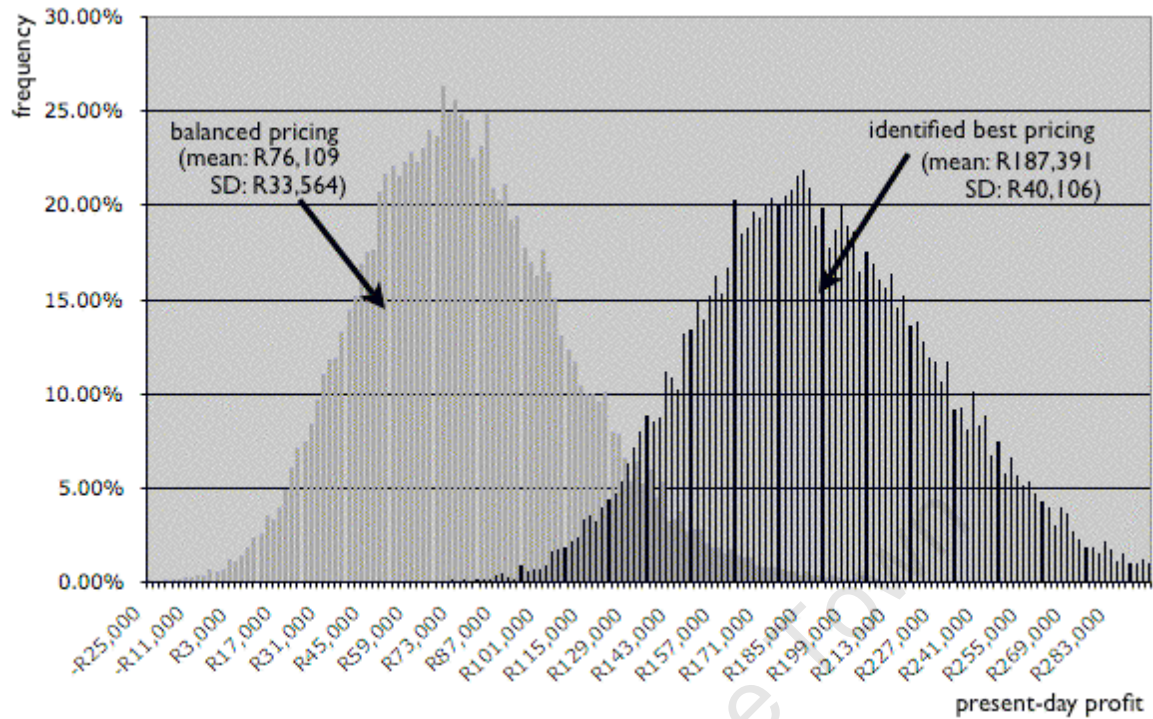


Figure 9.6 Present-day value of profit, showing the benefit of the pricing identified by the model

Figure 9.7 below shows the improvement on the contractor's cumulative nett cashflow using the recommended pricing, by comparison to the balanced pricing.

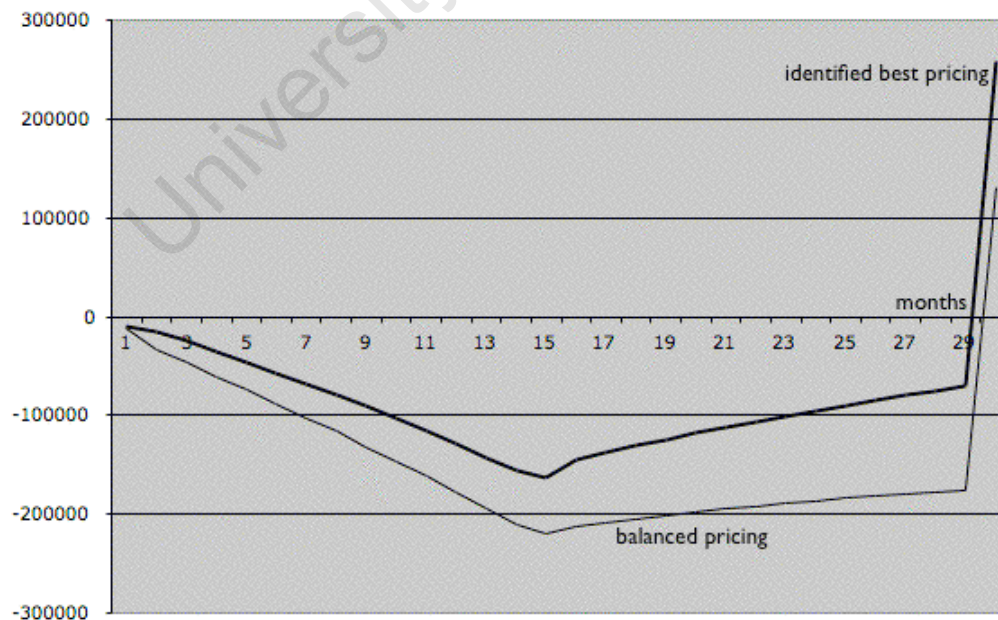


Figure 9.7 The effect of the unbalanced prices on the hypothetical project's nett cashflow (showing mean values)

The unbalanced pricing has also helped to improve the contractor's escalation compensation. Figure 9.8 shows that if the pricing were balanced, the contractor would receive a mean, expected R545,128 for escalation, whereas with the recommended pricing, this increases to R556,557.

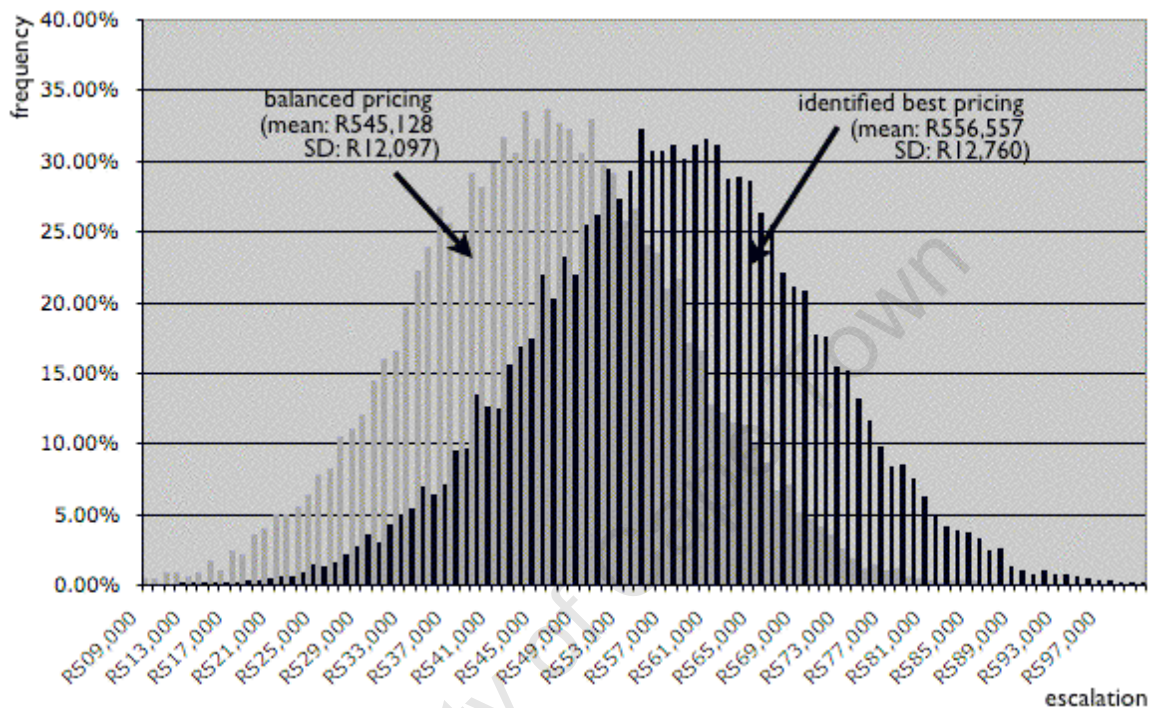


Figure 9.8 The compensation for escalation, showing the benefit of the pricing identified by the model

The pricing has also improved the contractor's value of likely variations. With balanced pricing, as it is shown in Figure 9.9 below, they would, most likely, be paid R3,450 less for reason of these variations (as anticipated by the contractor) whereas, with the recommended pricing, they will be paid (a mean of) R110,407 extra.

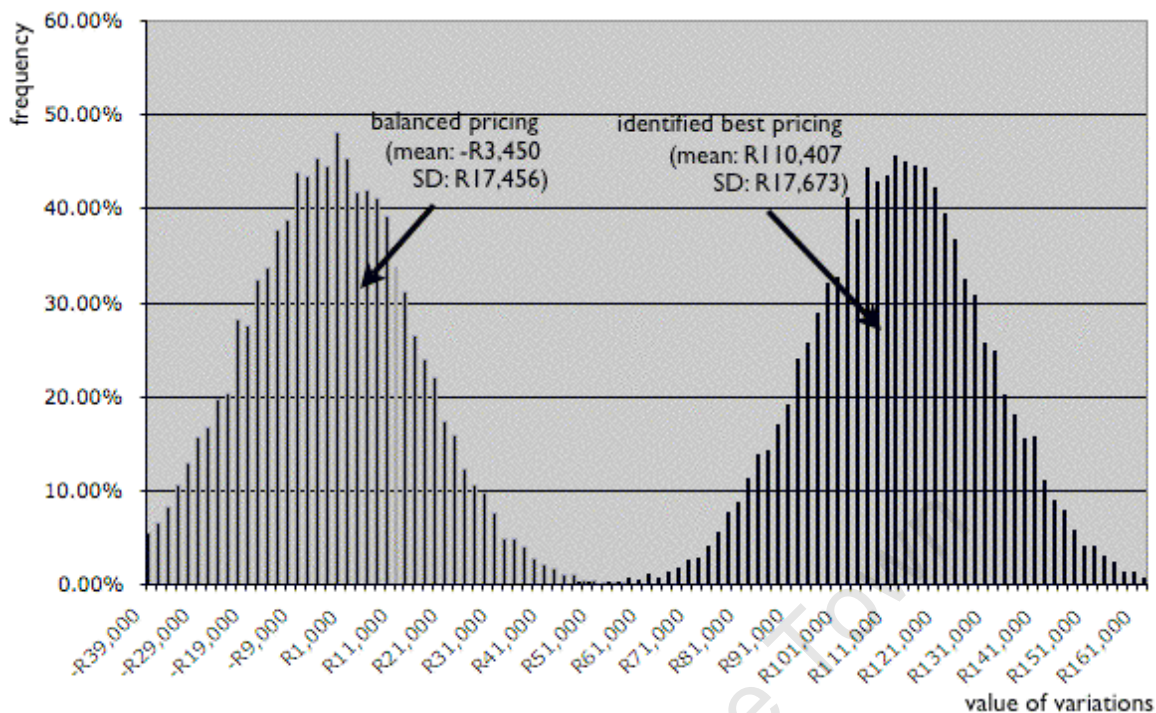


Figure 9.9 The compensation for contract variations, showing the benefit of the pricing identified by the model

Notice that, as observed in Chapter 6, more profit could be accomplished if the contractor were willing to accept more risk. In this instance, these prices reflect the compromise between profitability and risk that the contractor feels to be most satisfying to them.

Notice from the pricing listed in Appendix E that the system has managed to constrain the prices to within the minimum and maximum bounds provided as input by the contractor, with the result that the prices are likely to appear as relatively 'normal' by comparison to what the contractor has observed as being prevalent in the industry.

CONCLUSION

This test successfully provides validation of the application of the model. By using this system, in the case of the hypothetical test, the contractor enjoys a substantial increase in profitability without excessively exposing themselves to risk. Furthermore, the system gives contractors a methodology by which these dual, and opposing, objectives are managed in accordance with their assessed attitude to risk, presenting them with a balance between the two that would appear best suited to them. The system also gives

them the opportunity to review their risk profile and choose to accept more risk or seek to avoid such risk, and then be able to do so on a scientific basis.

The system is also of value at ensuring that projects can be priced on a consistent, scientific and professional basis, governed by managerial parameters that can ensure that the decision-making involved is reasoned and reliable.

This system also provides proof that ‘unbalanced bidding’ need not be excessive in order to be significantly valuable. As mentioned in Chapter 2, unbalanced bidding is often criticized for reason that it is felt that it amounts to an unethical abuse of the system by a contractor. These tests show that unbalanced bidding can be subtle, within the bounds of what is likely to be deemed ‘acceptable’ by all stakeholders, and yet still able to contribute substantial benefits for contractors. It is suggested that the nature of pricing that this model has identified in this instance, is unlikely to be perceived by any client as being excessively ‘loaded’.

The system that has been developed is only a prototype and could not be deployed, in its current state, in a practical, real-world environment. It does, however, represent a new paradigm for item pricing. It presents a conceptual framework – a prototype - for what could now be developed as a commercial system.

10. CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER WORK

This research has proposed a new approach to item pricing by building contractors. This approach makes use of pricing items in accordance with the market's expectation (as opposed to being in accordance with the contractor's costs), whilst pricing them to give effect to the maximum possible *value of utility*, taking account of both the rewards available from this practice as well as the risks that this generates.

The research problem was stated (Chapter 1) as:

Unbalanced bidding models have not addressed the risks of an uneven mark-up despite that this has, from the outset of research into this field 50 years ago, been identified as a significant consideration. These models have also not been comprehensive as regards quantifying the various sources of improved profits and so these have so-far failed to provide contractors with a meaningful or useful technique for optimising their item pricing with respect to profit and risk.

The research questions were identified as being the following:

- (1) *What are the benefits that can be derived from an uneven distribution of mark-up amongst a project's constituent items?*
- (2) *Is it possible that one model could comprehensively and collectively quantify all of these benefits?*
- (3) *What are the risks that contractor exposes themselves to, in the event that they price items without using a consistent mark-up?*
- (4) *Is a contractor able to quantify these risks, for any specific item price combination?*

- (5) *If it is possible for a contractor to quantify both the prospective profits as well as the risks for a comprehensive range of different item price combinations, how can a contractor choose from these to find the one best suited to them?*

The subsidiary questions necessary to inform these questions included:

- (1) *What other unbalanced bidding models have been proposed and how have these succeeded / failed to provide an effective solution?*
- (2) *What has research revealed that has been conducted in other areas of microeconomics as regards decisions that entail trading-off returns against risk?*
- (3) *What are the ethical considerations as regards whether it is acceptable for contractors to price items using different mark-ups and are they ethically obliged to price all items by way of using the same mark-up?*

The hypotheses tested by the research have been:

- (1) *The uneven allocation of mark-up between component items can have a significant effect on both the profitability and the risk of a project.*
- (2) *Some item price combinations may be considered more **efficient** than others: some such combinations will contribute the most expected profit for the same or lesser degree of risk than other combinations. On this basis, it is not rational for a contractor to choose to use any prices that are not **efficient**.*
- (3) *A contractor is able to identify the item prices that will deliver the best compromise for them between risk and reward, as judged in accordance with their personal attitude to risk.*

In this last chapter, the findings of the above questions are presented, followed by consideration of whether this research has proven the above hypotheses or not. Conclusions are then drawn from the research findings, and suggestions are then made for future research in this field. Finally, it is assessed whether the research succeeded as regards its aims and objectives.

10.1 THE FINDINGS

The findings with regard to the three **subsidiary questions** have been as follows:

(1) What other unbalanced bidding models have been proposed and how have these succeeded / failed to provide an effective solution?

Chapter 3 reviewed all of the literature in this field, over the past 50 years, since the beginning of research on unbalanced bidding. It identified that the various models that have been proposed all failed as regards...

- a. providing a comprehensive method of addressing all of the various rewards that unbalanced bidding can contribute;
- b. identifying the risks that are involved;
- c. measuring these risks; and
- d. providing any appropriate method by which a contractor can weigh-up the pursuit of more profit against the exposure to more risk

...thereby equipping contractors so that they are able to decide prices so as to accomplish a meaningful trade-off between these two objectives, if both of these are treated as being of equal importance.

(2) What has research revealed that has been conducted in other areas of microeconomics as regards decisions that entail trading-off returns against risk?

Modern Portfolio Theory (Chapter 6) has made a substantial contribution to change the manner in which investment portfolios are now chosen. This entails using a technique that identifies an Efficient Frontier subset of portfolios: those that offer more return, for the amount of risk that they generate, than any other.

Expected Utility Theory (Chapter 7), and its progression through to the more recently formulated Cumulative Prospect Theory, also provides a method of assessing any given scenario that entails a gain or loss, that is uncertain, or in other words that has some risk

attached. CPT provides a formulation whereby, if a person were to be interviewed so as to ascertain their attitude to matters of risk, a more general equation can be applied so as to identify their likely response to other matters that also involve risk.

Both of these fields provide useful frameworks that can be adopted by building contractors faced with choosing between different item price combinations that offer alternative rewards and risks.

(3) What are the ethical considerations as regards whether it is acceptable for contractors to price items using different mark-ups and are they ethically obliged to price all items by way of using the same mark-up?

It has been proposed (Chapter 2) that there is nothing that is ethically superior about bids that are “balanced”. Contractors have widely divergent costs and their estimates of these costs are even more divergent. In a hypothetical situation in which numerous contractors were to be submitting the same tender price, if these contractors were to all submit fully priced bills of quantities, a client may even, at times, have cause to prefer one of these that are “unbalanced” to another that is “balanced”. Depending on a contractor’s actual costs and their estimates of these, it is possible that an unbalanced set of prices may be more resemblant of the industry norm (and more acceptable to a client), than another contractor’s set of “balanced” prices. It has also been shown that unbalanced pricing does not equate to “extreme” pricing or extortionate pricing. For instance, the pricing that the test revealed in Chapter 9 is unlikely to be interpreted as extortionate and yet it is arguably highly unbalanced (mark-ups ranging from –9.0% to 38.4%, as shown in Appendix E).

The findings as regards the five **principle research questions** have been as follows:

(1) What are the benefits that can be derived from an uneven distribution of mark-up amongst a project’s constituent items?

Prior research has revealed these to be the prospective improvements to a contractor's cashflow, as well as the increased valuation they will receive for any variations that are expected in the quantities of certain items contained in the bills of quantities of the project. The current research has additionally identified a further benefit to be a prospective increase in the compensation for escalation the contractor will receive.

(2) Is it possible that one model could comprehensively and collectively quantify all of these?

A model has been formulated (Chapter 4) that does facilitate the measurement of the revenue that can be derived from each item, and from all items in summation, given a chosen price for each item. This model facilitates that prices can be identified to take advantage of a combined use of 'front-end loading', 'back-end loading' and 'quantity error exploitation'.

(3) What are the risks that contractor exposes themselves to, in the event that they price items without using a consistent mark-up?

The research (Chapter 5) has formulated these risks for the first time: describing them as the risks of 'rejection', 'reaction', and of 'being wrong'. Furthermore, it has found that these risks are of two different forms: which are described as being 'direct' and 'indirect' risks. Direct risks result from an item's price, whilst indirect risks are systemic and will arise regardless of a price (although indirect risks are proportional to prices). Direct risks are least when an item's price is most resemblant of the industry norm, whereas indirect risks are least when a price is *nil*.

(4) Is a contractor able to quantify these risks, for any specific item price combination?

A model has been formulated that uses a Value-at-Risk method of quantifying these risks and this has been presented in Chapter 5. This is the first model to identify and quantify the risks of item pricing. This model can measure the risk generated by any item, or all items in summation, given any item price, or combination of prices.

(5) *If it is possible for a contractor to quantify both the prospective profits as well as the risks for a comprehensive range of different item price combinations, how can a contractor choose from these to find the one best suited to them?*

This research has identified that it is useful to apply MPT (Chapter 6) and CPT (Chapter 7). These are techniques that were developed in the mainstream microeconomic arena, typically for the purposes of investment decision-making. These techniques can be applied to the domain of contractor's item pricing as the theoretical basis by which to single out one item pricing scenario that will deliver the greatest *value of utility*, taking account of both the expected profit as well as the risks of a range of alternative pricing combinations. A software system has been developed (Chapter 8) that implements and tests the new theory. This system uses Monte Carlo simulation, aided by a hybrid combination of artificial intelligence, genetic computing and fuzzy logic.

10.2 THE RESEARCH HYPOTHESES

The research has sought to provide proof that: (1) *the uneven allocation of mark-up between component items can have a significant effect on both the profitability and the risk of a project.* A test has been conducted on the hypothetical project described in Appendix B. A sample of the results from this test is shown in Appendix C, whilst Figures 9.2 and 9.3 have illustrated the considerable range of profit, risk and utility from different item price combinations.

The MPT method (described in Chapter 6), when combined with use of the models that have developed to measure the returns (Chapter 4) and the risks (Chapter 5) of item pricing, have provided proof that: (2) *some item price combinations may be considered more **efficient** than others: some such combinations will contribute the most expected profit for the same or lesser degree of risk than other combinations. On this basis, it is not rational for a contractor to choose to use any prices that are not **efficient**.*

The CPT method (described in Chapter 7), together with the system of implementation described in Chapter 8, have provided proof that (3) *a contractor is able to identify the*

item prices that will deliver the best compromise for them between risk and reward, as judged in accordance with their personal attitude to risk.

Chapter 9 shows the results of a test done (using the hypothetical project described in Appendix B), of which the highlights are as follows:

- a present-value profit of around 150% more from the identified pricing than from a set of ‘balanced’ prices, whilst keeping the risk within the acceptable bounds that were set,
- providing a basis by which more profit could be accomplished if the contractor were to indicate that they were more tolerant of risk, and conversely, providing the means to reign in the risk (albeit with some forfeiture of profit),
- an improvement to the cashflow to the contractor (as shown in Figure 9.7),
- an improvement to the escalation compensation that the contractor can expect to be paid (as shown in Figure 9.8),
- an improvement to the valuation of anticipated variations (as shown in Figure 9.9),
- pricing that appears ‘reasonable’ and ‘sensible’ (as shown in Appendix E) given the description of the project in Appendix B.

Furthermore, it is to be noted that the test results show pricing that has effectively become bound by limits that are more *fuzzy* in nature than fixed. The higher the risks that a contractor is willing to accept, the more *loaded* or *unbalanced* the pricing will become, giving effect to pricing that will be more removed from the pricing that would exist if the normal (*balanced*) average mark-up had instead been applied equally to all items. This theory is heuristically preferred relative to the previous status-quo in which unbalanced models were effectively affixing prices to exactly match either their (arbitrarily chosen) upper or lower pricing limits.

10.3 CONCLUSIONS

The research has produced a new theory and model for component item pricing. This new theory has managed to incorporate an assessment of the risks of item pricing, facilitating that equal emphasis can be given to both the risks as well as the returns in the pursuit of identifying a contractor's best-suited item price combination.

The new composite model has also achieved the effect that prices are bound by limits that are *fuzzy* in nature rather than fixed, and that a centrifugal nature of force acts upon these prices when a contractor is less concerned about risk; whilst, at other times when a contractor may be uncomfortable with the degree of risk, a centripetal nature of force serves to contain these prices and reign them in.

Part of the new theory is also that item prices should be referenced to the market's expectation of prices for such items, rather than referenced to the contractor's specific estimate of their cost.

The research has also shown that *unbalanced bidding* need not be considered as the equivalent of extortionate or extreme bidding. Instead, it is suggested that it is legitimate for contractors to use sophisticated methods of unbalanced bidding, in the same way that it has become common practice that they use sophisticated methods (such as the Critical Path Method) for project management for their scheduling of project activities. Indeed, it would appear that contractors who fail to use unbalanced bidding, else who fail to do so in a manner that is properly managed, with scientific precision, will be failing to accomplish much of the rewards that they can derive from projects, else that they will be exposing themselves to unnecessarily excessive risks. Competitive forces may drive all contractors to employ these methods.

10.4 RECOMMENDATIONS FOR FURTHER RESEARCH

There is considerable potential for further research in this field. The intention with this thesis has been to lay the foundation for a fresh perspective on unit pricing with the hope

that this perspective might lead to the further research that is necessary to evolve this science to the extent that it might become of practical value to contractors.

The many aspects that offer potential for further research include the following:

- a) Refining the model and the system, and in particular, optimising the code of the system so as to make it faster.
- b) Further consideration of the determination of the appropriate utility function and of the normalised scaling of the risks involved, including the measurement of any contractor's risk profile.
- c) Adapting the model so as to work with whatever data are available in the real world, looking to integrate them into construction computing systems, and testing the significance / importance of 'good' vs. 'bad' quality of data.
- d) Measuring the variability prevalent in pricing in different workgroups, serving to aid contractors with determining reasonable settings for the range of acceptable pricing for each item.
- e) Assessing the macroeconomic impact of the costs to clients of unbalanced bidding relative to the gains that are enjoyed by contractors. In particular, exploring the hypothesis that the improved survival of contractors may be a cost worth bearing by the industry.
- f) Exploring the hypothesis that a widespread prevalence of unbalanced bidding may effectively lead to lower costs to clients, as free-market forces drive contractors to pass on their benefits to clients, and as unbalanced bidding makes contractors more financially efficient (in terms of less dependent on capital) by comparison to using balanced pricing.
- g) Surveying the current-day prevalence of unbalanced bidding and modelling the macroeconomic costs to the industry of this practice.

- h) Pursuing the hypothesis that contractors might be forced to 'follow suit' – that they also adopt this practice - in any economic climate in which this practice has taken hold as being widespread.
- i) Adapting this system to other industries such as oil and forestry.
- j) Considering the use of Component Unit Pricing Theory as a marketing tool in the retail sector for the optimisation of strategies that entail differentiated mark-ups and the use of 'loss-leaders'.
- k) Exploring mathematical alternatives to the use of Monte Carlo simulation.
- l) Identifying alternative contractual arrangements that can circumvent the exposure to clients from unbalanced bidding.
- m) Finding effective quantitative methods for use by professional quantity surveyors for advising their clients on the extent and risks of unbalanced bids.
- n) Adding to the debate as regards the ethics of unbalanced bidding, and in particular, most likely, surveying those involved in the industry so as to assess the current perception of the prevalence of this practice and whether it is considered acceptable in ethical terms.
- o) Further investigating the ethical dilemma as regards unbalanced bidding and why it is that balanced bidding is often singled-out as being important in this regard, and yet little regard is had to the determination of the underlying base item 'costs' upon which these evenly-spread mark-ups are meant to be applied. In particular, investigating why 'balanced bids' are regarded as having a magical ethical property, despite that no two contractors' balanced bids will be the same.
- p) Pursuing the ethical paradox created when a contractor has an unusual exposure to costs and yet may be feeling compelled to submit a 'balanced bid'. If they submit a 'balanced bid' it might give the appearance of being unbalanced, and yet they will need to unbalance their bid to give the appearance of a balanced bid. Clients may, indeed, prefer their bid to be unbalanced than balanced in this case, which

highlights the paradox. Such situations could arise when contractors have abnormally high or low costs as might arise if they own protected proprietary rights to alternative materials, alternative sources of materials (such as a quarry) and / or methods of construction in which they have invested their capital. In other situations, they may unavoidably be exposed to excessive costs (relative to the industry norm) due to having to make use of inappropriate plant that they will have available at the time of construction. They might otherwise enjoy the benefit of less-than-normal costs if they are able to make use of old plant that has already been ‘written off’ their books.

- q) Consider the special case of subcontracting. It is hypothesised that subcontracting gives the main-contractor an extraordinary opportunity to reap the benefits of Component Unit Pricing Theory. In such instances, contractors have the subcontractors’ prices as their guide as regards market prices. Nevertheless, contractors can adjust these prices and apply an irregular mark-up before passing these prices on to their client.

10.5 RESEARCH AIMS AND OBJECTIVES

This research has *establish[ed] a scientific basis for more-effective, better-informed item pricing by contractors*, as was stated (Chapter 1) to be the aim. It has *(1) extend[ed] the present theoretical foundation for component item pricing, (2) in particular, [by way of] gain[ing] better insight into both the risks as well as the rewards generated from item pricing*. It has also *(3) establish[ed] a new mathematical model that quantifies both the risks and rewards of item pricing - to facilitate the identification of item prices that will give effect to a compromise between the pursuit of rewards, together with the restraint required to avoid excessive risk, as is suited to the circumstance and psychology of a particular contractor*. Suffice it to say that the aim of the research, together with its associated objectives, has been achieved.

APPENDIX A. SOFTWARE

This appendix provides the Java source code of the Xpload software that has been written to test of the model proposed in this thesis. This software has not been written with the purpose of functioning commercially. There has been little regard given to the user-interface or to the processing speed, for instance.

```
1 /**
2  * Xpload.java
3  * Copyright (c) 2009 - David Cattell, cattell@mac.com / dcattell@carwise.info
4  */
5
6 package oxbridge.carwise.enterprise;
7
8 import com.sun.org.apache.bcel.internal.verifier.statics.DOUBLE_Upper;
9 import java.util.Scanner;
10 import java.io.File;
11 import java.io.FileNotFoundException;
12 import java.io.FileOutputStream;
13 import java.io.FileDescriptor;
14 import java.text.*;
15
16 public class Xpload {
17     static int NUMBER_OF_ITEMS = 0;
```

```

18     static int NUMBER_OF_MONTHS = 30;
19     static final int NUMBER_OF_ASSOCIATES = 150;
20     static final int LOW = 0;
21     static final int MID = 1;
22     static final int HIGH = 2;
23     static final double eps = 1.0 + 1.0e-5;
24     static final boolean splitNORMALdistribution = true;
25     static int[][] assItem;
26     static int[][] assLowItem;
27     static final int lossFromRejection = 100000; // the loss the contractor will incur if
their pricing leads to the client rejecting their bid
28     static int[] maxProjectRisk = new int[2];
29     static int[] minProjectRisk = new int[2];
30     static int[][][] maxItemRisk = new int[504][2][2];
31     static int[][][] minItemRisk = new int[504][2][2];
32
33     /** Creates a new instance of Xpload */
34     public Xpload() {
35     }
36
37     static public void main(String[] args) {
38         final float discountingRatePerc = 10F; // i.e. use the discounting rate of 10%
39         double rj = (Math.pow((double)discountingRatePerc/100.0 + 1.0,1.0/12.0))-1.0;
40         double discFactor = (1.0/(1.0+rj));
41         double discSD = (1.0/(1.0+ 0.2 * rj)) - discFactor; // where 0.2 shows
that we're assuming that the standard deviation on the contractor's estimate of the
appropriate discounting rate is 20% of the rate he estimates
42         double[] escFactor = new double[3];
43         float HAYLETT_FACTOR = 0.85F;
44         float WINNERS_CURSE = 0.03F; // let us assume that the contractor's costs
are underestimated by 3% on average, seeing as they won the tender with the lowest price
(suggesting that the odds are that their cost estimate has been low)

```

```

45     {
46         double[] esc = {1, 12, 12}; // i.e. LOW = 1%(High confidence)      HIGH =
12%(Low confidence)      HIGH = 12%(High confidence)
47         for(int i = 0; i < esc.length; i++) {
48             escFactor[i] = Math.pow((double)esc[i]/100.0 + 1.0,1.0/12.0);
49         }
50     }
51     double[] escSD = {(escFactor[0] - 1.0) * 0.10, (escFactor[1] - 1.0) * 0.30,
(escFactor[2] - 1.0) * 0.10}; // reflecting that it is being assumed that the contractor
is much more confident of the escalation prediction for the 1st and 3rd items than the
2nd one
52     float RETENTION = 0.10F;
53     float RETENTION_CAP = 0.05F;
54
55     String dir = "/PhD_data/";
56     String itemFileName = dir + "items.txt";
57     String scheduleFileName = dir + "schedule.txt";
58     numberOfItems(itemFileName);
59     float[] itemCost = new float[NUMBER_OF_ITEMS];
60     float[] origLoPriceGuide = new float[NUMBER_OF_ITEMS];
61     float[] origHiPriceGuide = new float[NUMBER_OF_ITEMS];
62     int[] billQuantity = new int[NUMBER_OF_ITEMS];
63     float[] itemPrice = new float[NUMBER_OF_ITEMS];
64     float[] itemPriceSD = new float[NUMBER_OF_ITEMS];
65     float[] assignedMinPrice = new float[NUMBER_OF_ITEMS];
66     float[] assignedMaxPrice = new float[NUMBER_OF_ITEMS];
67     float[] expectedVariation = new float[NUMBER_OF_ITEMS];
68     float[] varSD = new float[NUMBER_OF_ITEMS];
69     int[] workGroup = new int[NUMBER_OF_ITEMS];
70     float[] rateRetention = new float[NUMBER_OF_MONTHS];
71     float[][] lambda = new float[NUMBER_OF_ITEMS][NUMBER_OF_MONTHS];
72     boolean[] confidentOfScheduledTiming = new boolean[NUMBER_OF_ITEMS];

```

```

73     float[] minimumPricingCommitment = new float[NUMBER_OF_ITEMS];
74     float[] maximumPricingCommitment = new float[NUMBER_OF_ITEMS];
75     assItem = new int[NUMBER_OF_ITEMS][NUMBER_OF_ASSOCIATES];
76     assLowItem = new int[NUMBER_OF_ITEMS][NUMBER_OF_ASSOCIATES];
77     float tenderPrice = 0;
78     final int NUMBER_OF_ITERATIONS = 2000000;
79     double bestResult = 0;
80     double bestRecentResult = 0;
81     int[] pricingSequence = new int[NUMBER_OF_ITEMS];
82     int[][] bestPricingSequences = new int[NUMBER_OF_ITEMS][3];
83     double[][][] bestItemUtility = new double[2][NUMBER_OF_ITEMS][10];
84     double[][][] bestInvItemUtility = new double[2][NUMBER_OF_ITEMS][10];
85     float[][][] bestItemPrices = new float[2][NUMBER_OF_ITEMS][10];
86     float[][][] bestItemPriceSD = new float[2][NUMBER_OF_ITEMS][10];
87     float[][][] bestLowItemPrices = new float[2][NUMBER_OF_ITEMS][10];
88     float[][][] bestLowItemPriceSD = new float[2][NUMBER_OF_ITEMS][10];
89     for (int item = 0; item < NUMBER_OF_ITEMS; item++) {
90         pricingSequence[item] = item; // to start with, before we reshuffle
them
91     }
92     int randomNumber, item, i, j;
93     float randomPrice;
94     float assignedPricing;
95
96     int[] pricingScheduleReward = new int[NUMBER_OF_ITERATIONS];
97     int[] pricingScheduleRisk = new int[NUMBER_OF_ITERATIONS];
98     float[] pricingScheduleUtility = new float[NUMBER_OF_ITERATIONS];
99     float[][] pricingSchedule = new float[NUMBER_OF_ITEMS][10];
100    float[][] pricingScheduleSD = new float[NUMBER_OF_ITEMS][10];
101    int[] orderResults = new int[NUMBER_OF_ITERATIONS];
102    int[] orderItemResults = new int[NUMBER_OF_ITEMS];
103    int[] itemRanking = new int[NUMBER_OF_ITEMS];

```

```

104     int iteration = 0;
105     int GENERATION = 1000;
106     boolean firstGeneration = true;
107     boolean beyondFirstGeneration = false;
108     boolean birthOfNextGeneration = false;
109     boolean birthOfSecondGeneration = false;
110     boolean deathOfGeneration = false;
111     double[] maxItemReward = new double[NUMBER_OF_ITEMS];
112     double[] itemBeta = new double[NUMBER_OF_ITEMS];
113     double[] balancedItemReward = new double[NUMBER_OF_ITEMS];
114     int[] balancedItemRisk = new int[NUMBER_OF_ITEMS];
115     double balancedBidReward = 0;
116     int balancedBidRisk = 0;
117     int FIRST_TEST = 0;
118     int SECOND_TEST = 1;
119     int THIRD_TEST = 2;
120     int REBOOT_RISK_SCALE = 20000;
121     boolean RESCALE_RISKS = true;
122     boolean RESCALED = false;
123     final boolean MSR_RISK_INCREMENTALLY = false;
124     double pvActualCost = 0;
125
126     // read input data from disk
127     {
128         float inputData[][] = new float[NUMBER_OF_ITEMS][8+NUMBER_OF_MONTHS+1];
129         inputData = readFile(itemFileName);
130         float total = 0;
131         for(i = 0; i < NUMBER_OF_ITEMS; i++) {
132             int c = 0;
133             total = 0;
134             confidentOfScheduledTiming[i] = false;
135             j = (int)inputData[i][c++];

```

```

136         billQuantity[j] = (int)inputData[i][c++];
137         expectedVariation[j] = inputData[i][c++];
138         varSD[j] = inputData[i][c++];
139         workGroup[j] = (int)inputData[i][c++];
140         itemCost[j] = inputData[i][c++];
141         origLoPriceGuide[j] = inputData[i][c++];
142         origHiPriceGuide[j] = inputData[i][c];
143         if(inputData[i][8] > 0.1)
144             confidentOfScheduledTiming[i] = true;
145         for (int month = 0; month < NUMBER_OF_MONTHS; month++) {
146             lambda[i][month] = inputData[i][8+month+1];
147             total += lambda[i][month];
148         }
149         if(total > 1.0 * eps || total < 1.0 / eps) {
150             double adj = 1.0 / total;
151             for (int month = 0; month < NUMBER_OF_MONTHS; month++) {
152                 lambda[i][month] *= adj;
153             }
154         }
155     }
156 }
157
158 // calculate the tender price, simply as a derivation of the
origPriceGuide[] values
159     double estimatedProjectCost = 0;
160     for (item = 0; item < NUMBER_OF_ITEMS; item++) {
161         tenderPrice += (origLoPriceGuide[item] + origHiPriceGuide[item]) / 2.0 *
billQuantity[item];
162         estimatedProjectCost += itemCost[item] * billQuantity[item];
163     }
164     double avgMarkUp = tenderPrice / estimatedProjectCost - 1;
165

```

```

166     float min, max;
167     boolean assessedWrongRisk = false;
168     int NUMBER_OF_TESTS = (NUMBER_OF_ITERATIONS > 10000 ? 10000 :
NUMBER_OF_ITERATIONS);
169     double[] wrongMean = new double[NUMBER_OF_ITEMS];
170     double[] wrongSD = new double[NUMBER_OF_ITEMS];
171
172     for(j = 0; j < NUMBER_OF_ITERATIONS; j++)
173         orderResults[j] = j;
174     for(j = 0; j < NUMBER_OF_ITEMS; j++)
175         orderItemResults[j] = j;
176
177     // estimate the rate of retention for each month - which is adequate to test
the efficacy of the model
178     float retainedFunds = 0;
179     boolean retentionStillInContention = true;
180     for (int month = 0; month < NUMBER_OF_MONTHS; month++) {
181         rateRetention[month] = RETENTION;
182         float monthsRetention = 0;
183         float monthsValuation = 0;
184         for (item = 0; item < NUMBER_OF_ITEMS; item++) {
185             monthsValuation += lambda[item][month] * billQuantity[item] *
expectedVariation[item] * Math.pow(((escFactor[workGroup[item]] - 1) * HAYLETT_FACTOR +
1),month+1) * (origLoPriceGuide[item] + origHiPriceGuide[item]) / 2.0;
186             monthsRetention = rateRetention[month] * monthsValuation;
187         }
188         if(retainedFunds + monthsRetention > RETENTION_CAP * tenderPrice &&
retentionStillInContention) {
189             monthsRetention = RETENTION_CAP * tenderPrice - retainedFunds;
190             rateRetention[month] = monthsRetention / monthsValuation;
191             retentionStillInContention = false;
192         } else {

```

```

193         if(!retentionStillInContention) {
194             rateRetention[month] = 0;
195             monthsRetention = 0;
196         }
197     }
198     retainedFunds += monthsRetention;
199 }
200
201     // now that we have an estimate of the rateRetention[] values...
202     System.out.println("About to start simulation (using "+NUMBER_OF_TESTS+"
iterations) to assess the risk of being wrong");
203     // do a Monte-Carlo simulation to assess the risk of being wrong
204     for (item = 0; item < NUMBER_OF_ITEMS; item++) {
205         double[] wrongResults = new double[NUMBER_OF_TESTS];
206         for (i = 0; i < NUMBER_OF_TESTS; i++) {
207             // 1. the contractor could be wrong as regards their estimate of the
appropriate discounting rate...
208             double simulatedDiscFactor = normrand(discFactor, discSD);
209
210             // 2. also when estimating the variance in item's quantity...
211             double simulatedExpectedVariation = normrand(expectedVariation[item],
varSD[item]);
212
213             // 3. also when estimating each workgroup's escalation rate...
214             double simulatedEscFactor = normrand(escFactor[workGroup[item]],
escSD[workGroup[item]]);
215
216             // 4. also with their scheduled timing of the project...
217             float[] simulatedLambda = simulateLambda(lambda[item],
confidentOfScheduledTiming[item]);
218
219             for (int month = 0; month < NUMBER_OF_MONTHS; month++) {

```

```

220         wrongResults[i] += Math.pow(simulatedDiscFactor, month+1) *
simulatedLambda[month] * billQuantity[item] * simulatedExpectedVariation *
221         Math.pow(((simulatedEscFactor - 1) * HAYLETT_FACTOR +
1),month+1) * (1.0 - rateRetention[month]));
222     }
223 }
224 // determine the mean and SD of the wrongResults[]
225 wrongMean[item] = mean(wrongResults);
226 wrongSD[item] = stdev(wrongResults);
227 }
228 assessedWrongRisk = true;
229 System.out.println("Finished simulation to assess the risk of being wrong");
230 float alpha = 0;
231 float beta = 0;
232 float lastUsefulBeta = 0, lastUsefulAlpha = 0;
233 boolean recentImprovement = false, notSoRecentImprovement = false,
nowStretchingAlpha = false, nowStretchingBeta = false, lastLeg = false;
234 int maxItemIndexTestedThusFar = 0;
235 float[] stillLearning = {0, 0, 0};
236 boolean invalidResult = false;
237
238 String pickDebug = "";
239 String[] pickDebugAll = new String[NUMBER_OF_ITEMS];
240 boolean ranked = false;
241 while (iteration < NUMBER_OF_ITERATIONS && (iteration < stillLearning[2] +
100000 || recentImprovement || notSoRecentImprovement ||
242         beta > lastUsefulBeta + 0.2 ||
243         alpha > lastUsefulAlpha + 0.2)) { // pack up when it appears that
the system is no longer still learning anything
244         if(iteration == REBOOT_RISK_SCALE && RESCALE_RISKS && !RESCALED) { //
then reboot
245             bestResult = 0;

```

```

246         bestRecentResult = 0;
247         System.out.println();
248         System.out.println("Switched to new scale");
249         RESCALED = true; // so that it won't trip up again when it reaches
this stage
250         iteration = THIRD_TEST; // 3; // to go back to the beginning, with
some acquired knowledge though remaining intact
251     }
252     firstGeneration = iteration < GENERATION && !RESCALED;
253     beyondFirstGeneration = !firstGeneration;
254     birthOfNextGeneration = iteration % GENERATION == 0;
255     deathOfGeneration = (iteration+1) % GENERATION == 0;
256     birthOfSecondGeneration = iteration == GENERATION;
257     int k = RESCALED ? 1 : 0;
258     double s = Math.random();
259     int kk = RESCALED ? (iteration < GENERATION ? 0 : (iteration > 1000 ? 1 :
s < (double)iteration / 1000.0 ? 1 : 0)) : 0; // switch across increasingly to the new-
found intelligence on the best pricing, fading them in over the course of 1000 iterations
260     assignedPricing = 0;
261     String sequence;
262
263     if(beyondFirstGeneration && iteration % 3 == 0) {
264         double seed = Math.random();
265         int oneOfTen = (int)(seed * 10);
266         sequence = "A" + oneOfTen;
267         for (item = 0; item < NUMBER_OF_ITEMS; item++)
268             pricingSequence[item] = bestPricingSequences[item][oneOfTen];
// rank them starting from the one that produces the greatest utility / price
269     } else if (iteration % 3 == 1) {
270         sequence = "B";
271         // reshuffle the sequence of priority for pricing all items
272         for (item = 0; item < NUMBER_OF_ITEMS; item++) {

```

```

273         randomNumber = item +
(int)Math.floor(Math.random()*(NUMBER_OF_ITEMS - item));
274         int rand = pricingSequence[randomNumber];
275         for(j = randomNumber; j > item; j--) {
276             pricingSequence[j] = pricingSequence[j-1];
277         }
278         pricingSequence[item] = rand;
279     }
280 } else {
281     // price an initial portion of the items in accordance with
orderItemResults and then randomly scramble the rest
282     int extent = (int)((NUMBER_OF_ITEMS-1) * Math.random());
283     sequence = "C" + extent;
284     for (item = 0; item < NUMBER_OF_ITEMS; item++)
285         pricingSequence[item] = orderItemResults[item%2 == 0 ? item / 2
: (NUMBER_OF_ITEMS-1) - (item / 2)]; // rank them starting from the one that
produces the greatest utility / price
286     for (item = extent; item < NUMBER_OF_ITEMS; item++) {
287         randomNumber = item +
(int)Math.floor(Math.random()*(NUMBER_OF_ITEMS - item));
288         int rand = pricingSequence[randomNumber];
289         for(j = randomNumber; j > item; j--) {
290             pricingSequence[j] = pricingSequence[j-1];
291         }
292         pricingSequence[item] = rand;
293     }
294 }
295
296     // plan ahead and determine the commitments that lie ahead which will
later serve to ensure that the item pricing will not conflict with the tender price
297     float mult = 0.0F;
298     do {

```

```

299         for (int ii = NUMBER_OF_ITEMS-1; ii >= 0; ii--) {
300             assignedMinPrice[pricingSequence[ii]] = (beyondFirstGeneration ?
(bestLowItemPrices[kk][pricingSequence[ii]][0] - mult *
bestLowItemPriceSD[kk][pricingSequence[ii]][0]) : origLoPriceGuide[pricingSequence[ii]]);
301             assignedMaxPrice[pricingSequence[ii]] = (beyondFirstGeneration ?
(bestItemPrices[kk][pricingSequence[ii]][0] + mult *
bestItemPriceSD[kk][pricingSequence[ii]][0]) : origHiPriceGuide[pricingSequence[ii]]);
302
303             minimumPricingCommitment[ii] = (ii < NUMBER_OF_ITEMS-1 ?
minimumPricingCommitment[ii+1] : 0)
304             + assignedMinPrice[pricingSequence[ii]] *
billQuantity[pricingSequence[ii]];
305             maximumPricingCommitment[ii] = (ii < NUMBER_OF_ITEMS-1 ?
maximumPricingCommitment[ii+1] : 0)
306             + assignedMaxPrice[pricingSequence[ii]] *
billQuantity[pricingSequence[ii]];
307         }
308         if(mult == 0)
309             mult = 0.1F;
310         else
311             mult *= 1.1;
312     } while(beyondFirstGeneration && (minimumPricingCommitment[0] >
tenderPrice || maximumPricingCommitment[0] < tenderPrice));
313     boolean priceAllRemainingItemsWithMinimumPrice = false;
314     boolean priceAllRemainingItemsWithMaximumPrice = false;
315     int ireset = (int)(Math.random() * (NUMBER_OF_ITEMS-1));
316     int preset = (int)(Math.random() * (NUMBER_OF_ITEMS-1));
317     for (int ii = 0; ii < NUMBER_OF_ITEMS; ii++) { // price all items in order
of this iteration's sequence
318         item = pricingSequence[ii];
319         if(priceAllRemainingItemsWithMinimumPrice) {
320             itemPrice[item] = roundCents(beyondFirstGeneration ?

```

```

assignedMinPrice[item] : origLoPriceGuide[item]);
321     itemPriceSD[item] = 0;
322     } else if(priceAllRemainingItemsWithMaximumPrice) {
323         itemPrice[item] = roundCents(beyondFirstGeneration ?
assignedMaxPrice[item] : origHiPriceGuide[item]);
324         itemPriceSD[item] = 0;
325     } else {
326         if(deathOfGeneration) {
327             randomPrice = bestItemPrices[kk][item][0]; // give it the
price that we've found (looking at this item in isolation) produces the highest utility
328             itemPriceSD[item] = 0;
329         } else {
330             if(firstGeneration) {
331                 randomPrice = (float)normrand((origHiPriceGuide[item] +
origLoPriceGuide[item]) / 2.0, (origHiPriceGuide[item] - origLoPriceGuide[item]) / 2.0);
332                 itemPriceSD[item] = (float)((origHiPriceGuide[item] -
origLoPriceGuide[item]) / 2.0);
333             } else {
334                 // Analyse the ten best prices so far for this item
(taken in isolation), to serve to guide the process going forward
335                 // If we are confident that a high price suits this item
then try one higher than before...
336                 float guidePrice, guideSD;
337                 boolean bestProducersOfUtility = itemRanking[item] <
(float)NUMBER_OF_ITEMS / 3F;
338                 boolean worstProducersOfUtility = itemRanking[item] >
(float)NUMBER_OF_ITEMS * 2F/3F;
339                 int individualItemProducesHighUtility = 0,
itemPriceInTopPerformingProjectsHigh = 0, bestItemsProducingUtility = 0, combo1 = 0,
combo2 = 0, combo3 = 0, combo4 = 0, individualItemProducesLowUtility = 0,
itemPriceInTopPerformingProjectsLow = 0, worstItemsProducingUtility = 0, combo5 = 0,
combo6 = 0, combo7 = 0, combo8 = 0;

```



```

each other
362                                     assLowItem[ai][lowAssociate] = aj;
363                                     lowAssociate++;
364                                     }
365                                 }
366                             }
367                             if(beyondFirstGeneration) {
368                                 // Let the next generation learn from its
predecessors.
369                                 // The next generation of iterations will be
seeded on the basis that this has learnt from
370                                 // the successes of past generations.
371                                 // "Successes" are not only defined by way of
item price combinations that yield the best overall utility
372                                 // but also this system is constantly on the
lookout for any cases of where any individual item's
373                                 // price (taken in isolation) is found to
contribute to an extraordinarily high utility.
374                                 // The genetic algorithm purposefully steers the
testing towards more and more unexpected situations
375                                 // in the hope that some gems of knowledge will
be acquired that will contribute later when evolution
376                                 // leads the analysis back to where it has found
the most likely domain for finding the best utility.
377                                 // It is therefore not of concern, but rather it
is representative of the intent, that the results go
378                                 // through a phase of getting worse before they
get better. The system learns from taking the risks
379                                 // whilst getting worse and takes to the extreme
until it finds no-more can be learnt from having strayed
380                                 // this far. It then heads back, armed with all
its knowledge that it has gained enroute, and

```

```

381 // finally seeks to find the best way by which
to take advantage of all that has been learnt.
382 // New settings for Beta and Alpha serve to
guide the Monte-Carlo Simulation process through the next
383 // generation of iterations.
384 // Beta determines the variation or "spread" of
the distribution of random numbers tested, whilst
385 // Alpha determines the relative value of their
mean compared to past tests. So, values for Alpha
386 // of anything other than 0.0 causes the new
tests to be shifted away from previous tests into a territory
387 // that might, to some extent, not have
previously been explored.
388 // High Beta values lend themselves to the
system making "wild guesses" whilst low Beta values are
389 // appropriate for when the system has honed its
knowledge to the extent that it may be more certain
390 // around where to find the best results.
391 // High Alpha values are used to explore new
ground far away from where previous tests have been done
392 // whilst low Alpha values lend themselves to
exploring the exact same territory as before.
393 String route = "a";
394 if(recentImprovement ||
395 ((beta >= (stillLearning[0] / eps) &&
alpha >= (stillLearning[1] / eps)) &&
396 (beta < (stillLearning[0]+0.2)*eps) ||
alpha < (stillLearning[1]+0.2)*eps))) {
397     route += "b";
398     if(recentImprovement) {
399         route += "c";
400         lastUsefulBeta = beta;

```

```

401                                     lastUsefulAlpha = alpha;
402                                     if(alpha > 0)
403                                         alpha -= 0.1F;
404                                     } else {
405                                         route += "d";
406                                         if(beta >= (stillLearning[0]+0.2)/eps &&
alpha >= (stillLearning[1]+0.2)/eps) {
407                                             route += "e";
408                                             if(beta > alpha)
409                                                 beta = 0;
410                                             else
411                                                 alpha = 0;
412                                         } else if(beta >=
(stillLearning[0]+0.2)/eps) {
413                                             beta = 0;
414                                             if(alpha >=
(stillLearning[1]+0.2)/eps)
415                                                 alpha = 0;
416                                             else
417                                                 alpha += 0.1F;
418                                             route += "f";
419                                         } else if(alpha >=
(stillLearning[1]+0.2)/eps) {
420                                             alpha = 0;
421                                             beta += 0.1;
422                                             route += "g";
423                                         } else {
424                                             beta += 0.1;
425                                             alpha += 0.1F;
426                                             route += "h";
427                                         }
428                                     }

```

```

429         } else if(notSoRecentImprovement) {
430             route += "i";
431             beta = 0;
432             alpha = 0;
433             notSoRecentImprovement = false; // reset
434         } else { // if we seem to be going nowhere, we now
need to embark on a new regimen by use to make good use of the intelligence that has
already been gathered
435             route += "j";
436             if(similar(beta, lastUsefulBeta) &&
similar(alpha, lastUsefulAlpha)) {
437                 route += "k";
438                 beta = 0.1F;
439                 alpha = 0.0F;
440             } else if (Math.random() < 0.5) {
441                 route += "l";
442                 beta = 0;
443                 alpha = 0;
444             } else {
445                 route += "l";
446                 beta = lastUsefulBeta;
447                 alpha = lastUsefulAlpha;
448             }
449         }
450         beta = roundOff(beta);
451         alpha = roundOff(alpha);
452         if(pickDebug.length() > 0) {
453             System.out.println();
454             System.out.println(pickDebug); pickDebug = "";
455         }
456         if(bestRecentResult != bestResult)
457             System.out.println("Best recent result =

```

```

"+bestRecentResult);
458             System.out.println(iteration+": Changed beta to
"+beta+" & alpha to "+alpha+" (having taken route "+route+"));
459             recentImprovement = false; // reset every
generation, regardless
460             bestRecentResult = 0;
461         }
462     }
463
464     boolean loadItem = false;
465     double direction = 0;
466     boolean reset = ii == 0 || ii == ireset; // gives effect
to a (possible) scrambling of the pickers with every new set of itemPrices
467     scramblePickMethod(reset);
468
469     if(pickMethod(itemBeta[item] > (mean(itemBeta) + 0.5 *
stdev(itemBeta)),
470             mean(pricingSchedule[item]) >
(origHiPriceGuide[item] + origLoPriceGuide[item]) / 2.0F,
471             bestProducersOfUtility)) { // load these
472         loadItem = true;
473         direction = +1;
474     } else if(pickMethod2(itemBeta[item] < mean(itemBeta),
475             mean(pricingSchedule[item]) <
(origHiPriceGuide[item] + origLoPriceGuide[item]) / 2.0F,
476             worstProducersOfUtility)) { // unload these
(i.e. drain these to provide funding for the loading)
477         loadItem = true;
478         direction = -1;
479     }
480
481     if(lastLeg) { // lastly focus on just trying to find the

```

```

best combination of the best item prices already found
482         scramblePicker(ii == 0 || ii == preset); // shake it up
483         boolean load = itemRanking[item] <=
(float)NUMBER_OF_ITEMS / 2F;
484         guidePrice = pickrand(load ?
bestItemPrices[kk][item][0] : bestLowItemPrices[kk][item][0]/* + direction * alpha *
bestLowItemPriceSD[item][0]*/,
485         load ?
bestItemPrices[kk][associatedItem(item)][0] : bestLowItemPrices[kk][associatedLowItem(item)][0],
486         pricingSchedule[item][0]/* + direction *
alpha * bestItemPriceSD[item][0]*/);
487         guideSD = 0;
488     } else {
489         scramblePicker(ii == 0 || ii == preset);
490         if(loadItem) {
491             int ra = (int)(Math.random() * (NUMBER_OF_ITEMS-
1));
492             if(direction > 0) {
493                 guidePrice =
pickrand(mean(bestItemPrices[kk][item]) + direction * alpha * bestItemPriceSD[kk][item][0],
494                 bestItemPrices[kk][item][0] +
direction * alpha * bestItemPriceSD[kk][item][0],
495                 bestItemPrices[kk][associatedItem(item)][0],
496                 bestItemPrices[kk][ra][0],
497                 pricingSchedule[item][0] + direction
* alpha * pricingScheduleSD[item][0],
498                 mean(pricingSchedule[item]) +
direction * alpha * pricingScheduleSD[item][0],
499                 origLoPriceGuide[item] +
(2.0+direction * alpha)/4.0 * (origHiPriceGuide[item] - origLoPriceGuide[item]),
500                 mean(bestItemPrices[kk][item])),

```

```

501                                     mean(pricingSchedule[item]),
502                                     (origHiPriceGuide[item] +
origLoPriceGuide[item]) / 2.0F);    // wild shot
503                                     guideSD = (float)(beta *
pick(bestItemPriceSD[kk][item][0] * 3,
504                                     bestItemPriceSD[kk][item][0],
505                                     bestItemPriceSD[kk][associatedItem(item)][0],
506                                     bestItemPriceSD[kk][ra][0],
507                                     pricingScheduleSD[item][0],
508                                     pricingScheduleSD[item][0],
509                                     (origHiPriceGuide[item] -
origLoPriceGuide[item]) / 3,
510                                     bestItemPriceSD[kk][item][0],
511                                     pricingScheduleSD[item][0],
512                                     (origHiPriceGuide[item] -
origLoPriceGuide[item]) / 2.0F));
513                                     } else {
514                                     guidePrice =
pickrand(mean(bestLowItemPrices[kk][item]) + direction * alpha *
bestLowItemPriceSD[kk][item][0],
515                                     bestLowItemPrices[kk][item][0] +
direction * alpha * bestLowItemPriceSD[kk][item][0],
516                                     bestLowItemPrices[kk][associatedLowItem(item)][0],
517                                     bestLowItemPrices[kk][ra][0],
518                                     pricingSchedule[item][0] + direction
* alpha * pricingScheduleSD[item][0],
519                                     mean(pricingSchedule[item]) +
direction * alpha * pricingScheduleSD[item][0],
520                                     origLoPriceGuide[item] +
(2.0+direction * alpha)/4.0 * (origHiPriceGuide[item] - origLoPriceGuide[item]),

```

```

521                                     mean(bestLowItemPrices[kk][item]),
522                                     mean(pricingSchedule[item]),
523                                     (origHiPriceGuide[item] +
origLoPriceGuide[item]) / 2.0F);    // wild shot
524                                     guideSD = (float)(beta *
pick(bestLowItemPriceSD[kk][item][0] * 3,
525                                     bestLowItemPriceSD[kk][item][0],
526                                     bestLowItemPriceSD[kk][associatedLowItem(item)][0],
527                                     bestLowItemPriceSD[kk][ra][0],
528                                     pricingScheduleSD[item][0],
529                                     pricingScheduleSD[item][0],
530                                     (origHiPriceGuide[item] -
origLoPriceGuide[item]) / 3,
531                                     bestItemPriceSD[kk][item][0],
532                                     pricingScheduleSD[item][0],
533                                     (origHiPriceGuide[item] -
origLoPriceGuide[item]) / 2.0F));
534                                     }
535                                     } else {    // until I can think of something better
to do with the others, perhaps just leave them at their original settings, barring
widening up the variance slightly (in an effort to increase the possibility of finding
something new)
536                                     guidePrice = pickrand(bestItemPrices[kk][item][0],
537                                     bestItemPrices[kk][associatedItem(item)][0],
538                                     bestLowItemPrices[kk][associatedLowItem(item)][0],
539                                     pricingSchedule[item][0],
540                                     mean(pricingSchedule[item]),
541                                     mean(bestItemPrices[kk][item]),
542                                     (origHiPriceGuide[item] +
origLoPriceGuide[item]) / 2.0F,

```

```

543                                     (origHiPriceGuide[item] +
origLoPriceGuide[item]) / 2.0F);    // wild shot
544                                     guideSD = (float)(beta *
pick(bestItemPriceSD[kk][item][0],
545
bestItemPriceSD[kk][associatedItem(item)][0],
546
bestLowItemPriceSD[kk][associatedLowItem(item)][0],
547                                     pricingScheduleSD[item][0],
548                                     pricingScheduleSD[item][0],
549                                     bestItemPriceSD[kk][item][0],
550                                     1.2F * (origHiPriceGuide[item] -
origLoPriceGuide[item]) / 2.0F,
551                                     (origHiPriceGuide[item] -
origLoPriceGuide[item]) / 2.0F));
552                                     }
553                                     }
554                                     pickDebugAll[item] = (direction > 0 ? "+" : direction <
0 ? "-" : " ") + stick;
555                                     itemPriceSD[item] = guideSD;
556                                     randomPrice = (float)normrand(guidePrice, guideSD);
557                                     if(randomPrice <= 0) {
558                                         if(guidePrice > 0)
559                                             randomPrice = guidePrice;
560                                         else
561                                             randomPrice = 0;
562                                     }
563                                     }
564                                     }
565                                     min = ii+1 < NUMBER_OF_ITEMS ? minimumPricingCommitment[ii+1] :
0;
566                                     max = ii+1 < NUMBER_OF_ITEMS ? maximumPricingCommitment[ii+1] :

```

```

0;
567         if(iteration == FIRST_TEST)
568             itemPrice[item] = origLoPriceGuide[item]; // to induce a
record of each item's max risk
569         else if(iteration == SECOND_TEST)
570             itemPrice[item] = origHiPriceGuide[item]; // to induce a
record of each item's max risk
571         else {
572             if(assignedPricing + billQuantity[item] * randomPrice <
tenderPrice - min &&
573                 assignedPricing + billQuantity[item] * randomPrice >
tenderPrice - max) {
574                 if(iteration == THIRD_TEST)
575                     itemPrice[item] = roundCents((float)(itemCost[item] *
(1.0 + avgMarkUp))); // to induce a record of the project's least risk
576                 else
577                     itemPrice[item] = roundCents(randomPrice);
578                 } else if(assignedPricing + billQuantity[item] * randomPrice <
tenderPrice - max) {
579                     itemPrice[item] = roundCents((tenderPrice - max -
assignedPricing) / billQuantity[item]);
580                     priceAllRemainingItemsWithMaximumPrice = true;
581                 } else {
582                     itemPrice[item] = roundCents((tenderPrice - min -
assignedPricing) / billQuantity[item]);
583                     priceAllRemainingItemsWithMinimumPrice = true;
584                 }
585             }
586         }
587         assignedPricing += itemPrice[item] * billQuantity[item];
588     }
589     if (iteration >= THIRD_TEST && (assignedPricing > tenderPrice * eps ||

```

```

assignedPricing < tenderPrice / eps)) {
590         System.out.println("ERROR: assignedPricing = "+assignedPricing+"
relative to tenderPrice = "+tenderPrice);
591         invalidResult = true;
592     }
593     double itemReward, reward = 0, riskRejection, riskReaction = 0,
riskOfBeingWrong = 0, result = 0, maxRiskRejection = 0;
594
595     if(!invalidResult) {
596         try {
597             String SEP = ", ";
598             String CR = "\r\n";
599             FileOutputStream flog = new FileOutputStream("dummy", true);
600             flog = openFlog(flog, dir + "inflowFactors.txt");
601             FileOutputStream flog2 = new FileOutputStream("dummy", true);
602             flog2 = openFlog(flog2, dir + "outflows.txt");
603             FileOutputStream flog3 = new FileOutputStream("dummy", true);
604             flog3 = openFlog(flog3, dir + "escalation.txt");
605             for (item = 0; item < NUMBER_OF_ITEMS; item++) { // evaluate the
effect of this pricing schedule
606                 String line = ""+item, line2 = ""+item;
607                 itemReward = 0;
608                 double itemRetention = 0, retention, monthsReward = 0,
monthsCost, itemsCost = 0, escalation = 0;
609                 for(int month = 0; month < NUMBER_OF_MONTHS; month++) {
610                     monthsReward = itemPrice[item] * lambda[item][month] *
billQuantity[item] * expectedVariation[item] * Math.pow(((escFactor[workGroup[item]] - 1)
* HAYLETT_FACTOR + 1),month+1);
611                     monthsCost = (itemCost[item] * (1.0 + WINNERS_CURSE)) *
lambda[item][month] * billQuantity[item] * expectedVariation[item] *
Math.pow(escFactor[workGroup[item]],month+1);
612                     retention = rateRetention[month] * monthsReward;

```

```

613             // If I wanted to add interest to the funds in the
retention fund, this is the place to add it
614             itemRetention += retention;
615             if(month < NUMBER_OF_MONTHS-1)
616                 monthsReward -= retention;
617             else
618                 monthsReward += itemRetention; // release the
retention from the fund in the last month
619             itemReward += Math.pow(discFactor, month+1) *
monthsReward;
620             itemsCost += Math.pow(discFactor, month+1) * monthsCost;
621             line += SEP + monthsReward / itemPrice[item];
622             line2 += SEP + monthsCost;
623             escalation += lambda[item][month] * billQuantity[item] *
expectedVariation[item] * (Math.pow(((escFactor[workGroup[item]] - 1) *
HAYLETT_FACTOR)+1,month+1) - 1);
624             }
625             line +=CR;
626             line2 +=CR;
627             String line3 = "" + item + SEP + escalation + CR;
628             if(iteration == FIRST_TEST) {
629                 writeLine(flog, line);
630                 writeLine(flog2, line2);
631                 writeLine(flog3, line3);
632             }
633             reward += itemReward;
634             itemBeta[item] = itemReward / (itemPrice[item] *
billQuantity[item]); // so each item's itemBeta should remain constant regardless of the
test price
635             float SD = (origHiPriceGuide[item] - origLoPriceGuide[item]) /
2F;
636             float origMeanPrice = (origHiPriceGuide[item] +

```

```

origLoPriceGuide[item]) / 2F;
637         double risk = normdist(itemPrice[item], origMeanPrice, SD);
638         double riskMax = normdist(origMeanPrice, origMeanPrice, SD); //
the least risk will arise at the origGuidePrice
639         riskRejection = (riskMax - risk) * (double)lossFromRejection /
riskMax; // to inverse it and standardize it
640         if(riskRejection > maxRiskRejection)
641             maxRiskRejection = riskRejection;
642         double lossFromReaction = itemsCost - itemReward;
643         if(iteration == 0 && !RESCALED) // only do it once
644             pvActualCost += itemsCost;
645         riskReaction += Math.abs((riskMax - risk) *
(double)lossFromReaction / riskMax);
646         riskOfBeingWrong += (wrongMean[item] - norminv(0.48,
wrongMean[item], wrongSD[item])) * itemPrice[item]; // where 0.3 is the probability
assigned to the VaR assessment
647         int iRisk = (int)(Math.abs((riskMax - risk) *
(double)lossFromReaction / riskMax) + (wrongMean[item] - norminv(0.48, wrongMean[item],
wrongSD[item])) * itemPrice[item] + riskRejection);
648         if(iteration <= THIRD_TEST) {
649             balancedItemReward[item] = itemReward;
650             balancedItemRisk[item] = iRisk;
651         }
652         double itemResult = utilityFunction(itemReward -
balancedItemReward[item], iRisk - (MSR_RISK_INCREMENTALLY ? balancedItemRisk[item] : 0),
item, !RESCALED, itemPrice[item] == 0, iteration == FIRST_TEST, iteration == SECOND_TEST,
iteration == THIRD_TEST); // (itemPrice[item] * billQuantity[item]);
653         if((Double.isNaN(maxItemReward[item]) || maxItemReward[item]
== 0) && itemPrice[item] > 0) {
654             float high = origHiPriceGuide[item]; // origMeanPrice + 2.0F
* (origHiPriceGuide[item] - origMeanPrice);
655             maxItemReward[item] = itemReward * high /

```

```

itemPrice[item];
656         }
657         double invItemResult = itemPrice[item] > 0 ?
utilityFunction(iteration > THIRD_TEST ? maxItemReward[item] - itemReward /*-
balancedItemReward[item]*/ : 0, iRisk - (MSR_RISK_INCREMENTALLY ? balancedItemRisk[item]
: 0), item, true, !RESCALED, itemPrice[item] == 0, iteration == FIRST_TEST, iteration ==
SECOND_TEST, iteration == THIRD_TEST) / (itemPrice[item] * billQuantity[item]) : 0;
658
659         if(item > maxItemIndexTestedThusFar && iteration >=
THIRD_TEST)
660             maxItemIndexTestedThusFar = item;
661             if(!Double.isNaN(invItemResult) && invItemResult != 0 &&
(invItemResult > bestInvItemUtility[k][item][9] || bestLowItemPrices[k][item][9] == 0)) {
662                 boolean knownAlready = false, bubbleUp = false,
betterThanBefore = false;
663                 int jj = 0;
664                 while(jj < 10 && bestLowItemPrices[k][item][jj] > 0 &&
!knownAlready) {
665                     if(itemPrice[item] == bestLowItemPrices[k][item][jj])
666                         knownAlready = true;
667                     jj++;
668                 }
669                 if(knownAlready && invItemResult >
bestInvItemUtility[k][item][jj-1])
670                     betterThanBefore = true;
671                 if((!knownAlready || betterThanBefore) && iteration >
THIRD_TEST) {
672                     int bubble = 9;
673                     while(bubble > 0 && ((invItemResult >
bestInvItemUtility[k][item][bubble-1]) || bestLowItemPrices[k][item][bubble-1] == 0)) {
674                         if(bestLowItemPrices[k][item][bubble] ==
itemPrice[item])

```

```

675         bubbleUp = true;
676         if(!knownAlready || bubbleUp) {
677             bestInvItemUtility[k][item][bubble] =
bestInvItemUtility[k][item][bubble-1];
678             bestLowItemPrices[k][item][bubble] =
bestLowItemPrices[k][item][bubble-1];
679             bestLowItemPriceSD[k][item][bubble] =
bestLowItemPriceSD[k][item][bubble-1];
680         }
681         bubble--;
682     }
683     bestInvItemUtility[k][item][bubble] = invItemResult;
684     if(beyondFirstGeneration &&
bestLowItemPrices[k][item][bubble] != itemPrice[item]) {
685         System.out.print("-"+bubble);
686         pickDebug += pickDebugAll[item];
687     }
688     if(bestLowItemPrices[k][item][bubble] !=
itemPrice[item] && stillLearning(bubble)) { // see the effects of moving along a lot faster
689         stillLearning[0] = beta; stillLearning[1] =
alpha; stillLearning[2] = iteration;}
690     bestLowItemPrices[k][item][bubble] =
itemPrice[item];
691     bestLowItemPriceSD[k][item][bubble] =
itemPriceSD[item];
692     }
693 }
694     if(!Double.isNaN(itemResult) && itemResult != 0 && (itemResult
> bestItemUtility[k][item][9] || bestItemPrices[k][item][9] == 0)) {
695         boolean knownAlready = false, bubbleUp = false,
betterThanBefore = false;
696         int jj = 0;

```

```

697         while(jj < 10 && bestItemPrices[k][item][jj] != 0 &&
!knownAlready) {
698             if(itemPrice[item] == bestItemPrices[k][item][jj])
699                 knownAlready = true;
700             jj++;
701         }
702         if(knownAlready && itemResult >
bestItemUtility[k][item][jj-1])
703             betterThanBefore = true;
704         if((!knownAlready || betterThanBefore) && iteration >
THIRD_TEST) {
705             int bubble = 9;
706             while(bubble > 0 && ((itemResult >
bestItemUtility[k][item][bubble-1]) || bestItemPrices[k][item][bubble-1] == 0)) {
707                 if(bestItemPrices[k][item][bubble] ==
itemPrice[item])
708                     bubbleUp = true;
709                 if(!knownAlready || bubbleUp) {
710                     bestItemUtility[k][item][bubble] =
bestItemUtility[k][item][bubble-1];
711                     bestItemPrices[k][item][bubble] =
bestItemPrices[k][item][bubble-1];
712                     bestItemPriceSD[k][item][bubble] =
bestItemPriceSD[k][item][bubble-1];
713                 }
714                 bubble--;
715             }
716             bestItemUtility[k][item][bubble] = itemResult;
717             if(beyondFirstGeneration &&
bestItemPrices[k][item][bubble] != itemPrice[item]) {
718                 System.out.print("+"+bubble);
719                 pickDebug += pickDebugAll[item];

```

```

720     }
721     if(bestItemPrices[k][item][bubble] != itemPrice[item]
&& stillLearning(bubble)) { // see the effects of moving along a lot faster
722         stillLearning[0] = beta; stillLearning[1] =
alpha; stillLearning[2] = iteration;}
723         bestItemPrices[k][item][bubble] = itemPrice[item];
724         bestItemPriceSD[k][item][bubble] =
itemPriceSD[item];
725     }
726 }
727
728 if(!ranked && iteration >= THIRD_TEST && !RESCALED) {
729     int oldPos = 0, newPos = 0;
730     for(i = maxItemIndexTestedThusFar; i >= 0; i--) {
731         if(itemBeta[item] >= itemBeta[orderItemResults[i]] ||
itemBeta[orderItemResults[i]] == 0)
732             newPos = i;
733         if(orderItemResults[i] == item)
734             oldPos = i;
735     }
736     if(newPos != oldPos) {
737         int add = 1;
738         if(newPos > oldPos)
739             add = -1;
740         i = oldPos;
741         while(i >= 0 && i < (maxItemIndexTestedThusFar+1)-(add
== 1 ? 0 : -1) && i != newPos) {
742             orderItemResults[i] = orderItemResults[i-add];
743             i -= add;
744         }
745         orderItemResults[newPos] = item;
746     }

```

```

747         if(maxItemIndexTestedThusFar == NUMBER_OF_ITEMS-1) {
748             for(i = 0; i < NUMBER_OF_ITEMS; i++)
749                 itemRanking[orderItemResults[i]] = i;
750         }
751     }
752 }
753     closeFlog(flog);
754 } catch(java.io.IOException ioe) {
755     ioe.printStackTrace();
756 }
757 if(iteration >= THIRD_TEST)
758     ranked = true;
759 if(iteration <= THIRD_TEST) {
760     balancedBidReward = reward;
761     balancedBidRisk = (int)(riskReaction + riskOfBeingWrong +
maxRiskRejection);
762 }
763     result = utilityFunction(reward - balancedBidReward, (int)(riskReaction +
riskOfBeingWrong + maxRiskRejection - (MSR_RISK_INCREMENTALLY ? balancedBidRisk : 0)),
!RESCALED, iteration == FIRST_TEST, iteration == SECOND_TEST, iteration == THIRD_TEST);
// in accordance with whatever the formula is
764
765     pricingScheduleReward[iteration] = Math.round(Math.round(reward));
766     pricingScheduleRisk[iteration] = Math.round(Math.round(riskReaction +
riskOfBeingWrong + maxRiskRejection));
767     pricingScheduleUtility[iteration] = (float)result;
768     if(result > bestResult || bestResult == 0) {
769         double prevBest = bestResult;
770         bestResult = result;
771         bestRecentResult = bestResult;
772         if(pickDebug.length() > 0) {
773             System.out.println();

```

```

774         System.out.println(pickDebug); pickDebug = "";
775     }
776     int improve = Math.round((float)((reward - balancedBidReward) /
(balancedBidReward - pvActualCost)) * 100F);
777     System.out.println("Improvement (due to item pricing priority
sequence " + sequence + ") to "+bestResult+" after "+iteration+" efforts, increasing the
reward to "+(int)reward+" (a "+improve+"% improvement on the profit from a balanced bid)
whilst reducing the total risk to "+(int)(riskReaction + riskOfBeingWrong +
maxRiskRejection)+"(riskReaction = "+(int)riskReaction+", riskOfBeingWrong =
"+(int)riskOfBeingWrong+" & riskRejection = "+(int)maxRiskRejection+)");
778     if(beyondFirstGeneration) {
779         if((int)Math.round(result) > (int)Math.round(prevBest)) {
780             recentImprovement = true;
781             notSoRecentImprovement = true;
782         }
783         if(!deathOfGeneration) {
784             stillLearning[0] = beta; stillLearning[1] = alpha;
785         }
786         stillLearning[2] = iteration;
787     }
788     } else {
789         if(result > bestRecentResult || bestRecentResult == 0)
790             bestRecentResult = result;
791     }
792
793     // Slot the new results into the position that it accomplishes in
the order of descending utility
794     int newPos = 0;
795     i = 0;
796     while(i <= iteration && (float)result <
pricingScheduleUtility[orderResults[i]]) {
797         newPos = i+1;

```

```

798         i++;
799     }
800     for(i = iteration; i > newPos; i--)
801         orderResults[i] = orderResults[i-1];
802     orderResults[newPos] = iteration;
803
804     if(newPos < 10) {
805         int bubble = 9;
806         while(bubble > 0 && ((result >
pricingScheduleUtility[orderResults[bubble-1]]) ||
pricingScheduleUtility[orderResults[bubble-1]] == 0)) {
807             for(i = 0; i < NUMBER_OF_ITEMS; i++) {
808                 pricingSchedule[i][bubble] = pricingSchedule[i][bubble-
1];
809                 pricingScheduleSD[i][bubble] =
pricingScheduleSD[i][bubble-1];
810                 if(newPos < 3 && bubble < 3)
811                     bestPricingSequences[i][bubble] =
bestPricingSequences[i][bubble-1];
812             }
813             bubble--;
814         }
815         for(i = 0; i < NUMBER_OF_ITEMS; i++) {
816             pricingSchedule[i][bubble] = itemPrice[i];
817             pricingScheduleSD[i][bubble] = itemPriceSD[i];
818             if(bubble < 3)
819                 bestPricingSequences[i][bubble] = pricingSequence[i];
820         }
821     }
822     invalidResult = false; // reset
823     iteration++;
824 }

```

```

825         if(iteration < 51000 ? iteration % 10000 == 0 : iteration % 50000 == 0)
826             writeResults(dir, iteration, bestItemPrices, bestItemUtility,
bestLowItemPrices, pricingSchedule, pricingScheduleReward, pricingScheduleRisk,
pricingScheduleUtility, orderResults);
827     }
828 }
829
830 static String customFormat(String pattern, double value) {
831     DecimalFormat myFormatter = new DecimalFormat(pattern);
832     String output = myFormatter.format(value);
833     return(output);
834 }
835
836 static String DecOneFormat(double value) {
837     return customFormat("#0.#", value);
838 }
839
840 private static void writeResults(String dir, int iteration, float[][][] bestItemPrices,
double[][][] bestItemUtility, float[][][] bestLowItemPrices, float[][] pricingSchedule, int[]
pricingScheduleReward, int[] pricingScheduleRisk, float[] pricingScheduleUtility, int[]
orderResults) {
841     try {
842         String SEP = ", ";
843         String CR = "\r\n";
844         FileOutputStream flog = new FileOutputStream("dummy", true);
845
846         // write the best item prices to disk
847         flog = openFlog(flog, dir + "bestPrices0hi_" + iteration + ".txt");
848         int it = 0;
849         String line = ""+it;
850         for(int i = 0; i < 10; i++)
851             line += SEP+bestItemPrices[1][it][i];

```

```

852     line +=CR;
853     writeLine(flog, line);
854
855     flog = openFlog(flog, dir + "bestPricesOutil_" + iteration + ".txt");
856     line = ""+it;
857     for(int i = 0; i < 10; i++)
858         line += SEP+DecOneFormat(bestItemUtility[1][it][i]);
859     line +=CR;
860     writeLine(flog, line);
861
862     // write the best item prices to disk
863     flog = openFlog(flog, dir + "bestPricesllo_" + iteration + ".txt");
864     for(int item = 0; item < NUMBER_OF_ITEMS; item++) { // loop through all
the items
865         line = ""+item;
866         for(int i = 0; i < 10; i++)
867             line += SEP+bestLowItemPrices[1][item][i];
868         line +=CR;
869         writeLine(flog, line);
870     }
871
872     // write the best schedule of prices to disk
873     flog = openFlog(flog, dir + "bestPrices_" + iteration + ".txt");
874     for(int item = 0; item < NUMBER_OF_ITEMS; item++) { // loop through all
the items
875         line = ""+item;
876         for(int sch = 0; sch < 10; sch++)
877             line += SEP+pricingSchedule[item][sch];
878         line +=CR;
879         writeLine(flog, line);
880     }
881

```

```

882         // write all the results from all of the tested alternative pricing
schedules
883         flog = openFlog(flog, dir + "bestResults_" + iteration + ".txt");
884         for(int res = 0; res < iteration; res++) // loop through all the results
885             writeLine(flog, ""+res+SEP+
886                 pricingScheduleReward[orderResults[res]]+SEP+
887                 pricingScheduleRisk[orderResults[res]]+SEP+
888                 pricingScheduleUtility[orderResults[res]]+CR);
889
890         // write all the results from all of the tested alternative pricing
schedules
891         flog = openFlog(flog, dir + "results_" + iteration + ".txt");
892         for(int res = 0; res < iteration; res++) // loop through all the results
893             writeLine(flog, ""+res+SEP+
894                 pricingScheduleReward[res]+SEP+
895                 pricingScheduleRisk[res]+SEP+
896                 pricingScheduleUtility[res]+CR);
897
898         closeFlog(flog);
899     } catch(java.io.IOException ioe) {
900         ioe.printStackTrace();
901     }
902 }
903
904 private static float roundCents(float price) {
905     int times100 = Math.round(price * 100);
906     return (float)times100 / 100F;
907 }
908
909 private static boolean stillLearning(int bubbles) {
910     if(bubbles == 0)
911         return true;

```

```

912     return false;
913 }
914 private static int associatedItem(int item) {
915     return assItem(item, assItem);
916 }
917 private static int associatedLowItem(int item) {
918     return assItem(item, assLowItem);
919 }
920 private static int assItem(int item, int aI[][]) {
921     int associatedItems = 0, i = 0, assocItem = item;
922     if(aI[0][0] > 0) {
923         while (associatedItems < NUMBER_OF_ASSOCIATES && aI[item][associatedItems]
< Integer.MAX_VALUE) {
924             associatedItems++;
925         }
926         i = (int)Math.round(Math.random() * associatedItems);
927         if(i == NUMBER_OF_ASSOCIATES)
928             i--;
929         assocItem = aI[item][i];
930     }
931     if(assocItem == Integer.MAX_VALUE) // there are no other items with a similar
price
932         assocItem = item;
933     return assocItem;
934 }
935
936 private static double utilityFunction(double reward, int risk, boolean preReset, boolean
loTest, boolean hiTest, boolean balTest) {
937     return utilityFunction(reward, risk, 999, false, preReset, false, loTest, hiTest,
balTest);
938 }
939

```

```

940     private static double utilityFunction(double reward, int risk, int item, boolean preReset,
boolean zeroPrice, boolean loTest, boolean hiTest, boolean balTest) {
941         return utilityFunction(reward, risk, item, false, preReset, zeroPrice, loTest,
hiTest, balTest);
942     }
943
944     private static double utilityFunction(double reward, int risk, int item, boolean low, boolean
preReset, boolean zeroPrice, boolean loTest, boolean hiTest, boolean balTest) {
945         boolean itemOnly = item <= NUMBER_OF_ITEMS;
946         int i = low ? 0 : 1;
947         int ii = preReset ? 0 : 1;
948         boolean postReset = !preReset;
949         if (item > NUMBER_OF_ITEMS) {
950             if(loTest || hiTest)
951                 maxProjectRisk[0] = risk;
952             if(balTest) {
953                 minProjectRisk[0] = risk;
954                 if(preReset) {
955                     minProjectRisk[1] = Integer.MAX_VALUE;
956                     maxProjectRisk[1] = Integer.MIN_VALUE;
957                 }
958             } else if(preReset) {
959                 if(risk < minProjectRisk[1])
960                     minProjectRisk[1] = risk;
961                 if(risk > maxProjectRisk[1])
962                     maxProjectRisk[1] = risk;
963             }
964         } else if(loTest || hiTest)
965             maxItemRisk[item][i][0] = risk;
966         else if(balTest) {
967             minItemRisk[item][i][0] = risk;
968             if(preReset) {

```

```

969         minItemRisk[item][i][1] = Integer.MAX_VALUE;
970         maxItemRisk[item][i][1] = Integer.MIN_VALUE;
971     }
972 } else if(preReset) {
973     if(risk < minItemRisk[item][i][1])
974         minItemRisk[item][i][1] = risk;
975     if(risk > maxItemRisk[item][i][1])
976         maxItemRisk[item][i][1] = risk;
977 }
978
979     int maxRisk = itemOnly ? maxItemRisk[item][i][ii] : maxProjectRisk[ii];
980     int minRisk = 0; // itemOnly ? minItemRisk[item][i][ii] : minProjectRisk[ii];
981     double riskRating = Math.min(1, Math.max(0, 1.0 - (double)(risk - minRisk) /
(maxRisk - minRisk))); // is 0 when the risk is greatest and 1 when the risk is least,
and linear inbetween
982     double weight = Wplus(riskRating);
983     if (reward < 0) // in the event of a loss
984         weight = 1 - Wminus(1 - riskRating);
985     return weight * util(reward);
986 }
987
988     private static double Wplus(double t) {
989         return weight(t, 0.61); // as per Tversky & Kahneman (1992)'s original
assessment
990     }
991
992     private static double Wminus(double t) {
993         return weight(t, 0.69); // as per Tversky & Kahneman (1992)'s original
assessment
994     }
995
996     private static double weight(double a, double b) {

```

```

997         return Math.pow(a, b) / Math.pow(Math.pow(a, b) + Math.pow(1.0-a, b), 1.0 /
b);
998     }
999
1000     private static double util(double y) {
1001         double CPT_alpha = 0.88;    // as per Tversky & Kahneman (1992)'s original
assessment
1002         double CPT_beta = 0.88;    // as per Tversky & Kahneman (1992)'s original
assessment, obviously potentially different from CPT_alpha
1003         double CPT_lambda = 2.25;  // ditto
1004         double lambda = 1.0;
1005         double factor = CPT_alpha;
1006         if (y == 0)
1007             return y;
1008         else if (y < 0) {
1009             factor = CPT_beta;
1010             lambda = -CPT_lambda;
1011         }
1012         double b = Math.abs(y);
1013         if (factor > 0)
1014             b = lambda * Math.pow(b, factor);
1015         else {
1016             if (CPT_beta == 0)
1017                 b = lambda * Math.log(b);
1018             else
1019                 b = lambda * (1 - Math.pow(b + 1.0, factor));
1020         }
1021         return b;
1022     }
1023
1024     private static float roundOff(float a) {
1025         int b = Math.round(a * 10);

```

```

1026     return (float)b / 10;
1027 }
1028
1029 private static boolean similar(double a, double b) {
1030     // to see if two values are materially the same, at the level of cents
1031     long inta = Math.round(a * 100); // to convert to cents
1032     long intb = Math.round(b * 100);
1033     if(a == b)
1034         return true;
1035     return false;
1036 }
1037
1038 private static float[] simulateLambda(float[] itemLambda, boolean confident) {
1039     float centralTime = 0;
1040     float simulatedcentralTime;
1041     float[] weeklySchedule = new float[itemLambda.length * 4];
1042     float[] newWeeklySchedule = new float[weeklySchedule.length];
1043     float[] simulatedLambda = new float[NUMBER_OF_MONTHS];
1044     int started = 0, ended = 0;
1045     boolean activityStarted = false;
1046     for(int month = 0; month < NUMBER_OF_MONTHS; month++) {
1047         centralTime += (month + 1) * itemLambda[month];
1048         for(int week = 0; week < 4; week++) { // assume a simple 4-week month,
each month
1049             int wk = month * 4 + week;
1050             weeklySchedule[wk] = itemLambda[month] / 4.0F; // assume a simple
even spread over the 4 weeks in the month
1051             if(!activityStarted && itemLambda[month] > 0) {
1052                 started = wk;
1053                 activityStarted = true;
1054             }
1055             if(itemLambda[month] > 0)

```

```

1056             ended = wk;
1057         }
1058     }
1059     if(confident)
1060         simulatedcentralTime = (float)normrand(centralTime, 0.05 * centralTime);
// a 5% SD being assumed for the timing of items where the contractor is confident of
their timing
1061     else
1062         simulatedcentralTime = (float)normrand(centralTime, 0.20 * centralTime);
// a 20% SD being assumed for the timing of items where the contractor is not confident
of their timing
1063
1064     float delay = (simulatedcentralTime - centralTime) * 4.0F;
1065     double sigma = 0;
1066     int firstActiveMonth = 0;
1067     int lastActiveMonth = 0;
1068     boolean alreadyActive = false;
1069     double potentialEffectOfOne = 0;
1070     double oldCentreOfGravity = 0;
1071     double interimNewCentreOfGravity = 0;
1072     started = Math.max(Math.round((float)started + delay), 0);
1073     ended = Math.min(Math.round((float)ended + delay), NUMBER_OF_MONTHS * 4 - 1);
1074     int month = 0;
1075     for(int week = 0; week <= ended; week++) {
1076         int corresponding = week;
1077         corresponding = Math.min(Math.max(Math.round((float)corresponding - delay),
0), weeklySchedule.length - 1); // constrained to avoid computing errors
1078         if(week >= started)
1079             newWeeklySchedule[week] = weeklySchedule[corresponding];
1080         simulatedLambda[month] += newWeeklySchedule[week];
1081         if((week+1) % 4 == 0 || week == ended) {
1082             oldCentreOfGravity += (1.0 + month) * itemLambda[month];

```

```

1083         interimNewCentreOfGravity += (1.0 + month) * simulatedLambda[month];
1084         sigma += simulatedLambda[month];
1085         if(simulatedLambda[month] > 0) {
1086             if(!alreadyActive) {
1087                 firstActiveMonth = month;
1088                 alreadyActive = true;
1089             }
1090             lastActiveMonth = month;
1091         }
1092         month++;
1093     }
1094 }
1095     potentialEffectOfOne = firstActiveMonth - lastActiveMonth; // the effect on
unadjusted of adding one unit to the first month of activity and taking this away from
the last month of activity
1096     double desiredNewCentreOfGravity = Math.min(Math.max(oldCentreOfGravity *
simulatedcentralTime / centralTime, 0), NUMBER_OF_MONTHS);
1097     double factorOfOne = (desiredNewCentreOfGravity - interimNewCentreOfGravity) /
potentialEffectOfOne;
1098     simulatedLambda[firstActiveMonth] += factorOfOne;           // addth a fraction
of the 'one'...
1099     simulatedLambda[lastActiveMonth] -= factorOfOne;           // ...and taketh it
away (to restore the balance)
1100     int i = firstActiveMonth;
1101     float kitty = 0;
1102     while(simulatedLambda[i] < 0) {
1103         kitty = -simulatedLambda[i];
1104         simulatedLambda[i] = 0;
1105         i++;
1106     }
1107     while(kitty > 0) {
1108         float tail = Math.min(kitty, simulatedLambda[i]);

```

```

1109         simulatedLambda[i] -= tail;
1110         kitty -= tail;
1111         i++;
1112     }
1113     float newSigma = 0;
1114     for(month = 0; month < NUMBER_OF_MONTHS; month++) {
1115         newSigma += simulatedLambda[month];
1116     }
1117     float newNewSigma = 0;
1118     boolean fix = false;
1119     for(month = 0; month < NUMBER_OF_MONTHS; month++) {
1120         simulatedLambda[month] /= newSigma;
1121         if(Float.isNaN(simulatedLambda[month]))
1122             fix = true;
1123         newNewSigma += simulatedLambda[month];
1124     }
1125     if(fix) {
1126         for(month = 0; month < NUMBER_OF_MONTHS; month++)
1127             simulatedLambda[month] = 0;
1128         if(desiredNewCentreOfGravity >= NUMBER_OF_MONTHS)
1129             simulatedLambda[NUMBER_OF_MONTHS-1] = 1;
1130         else
1131             simulatedLambda[0] = 1;
1132     }
1133
1134     if(newNewSigma > 1.0 * eps || newNewSigma < 1.0 / eps)
1135         System.out.println("ERROR: newNewSigma = "+newNewSigma);
1136
1137     for(i = 0; i < simulatedLambda.length; i++)
1138         if(Float.isNaN(simulatedLambda[i]))
1139             System.out.println("ERROR: simulatedAlpha["+i+"] =
"+simulatedLambda[i]);

```

```
1140
1141     return simulatedLambda;
1142 }
1143
1144 static double r = 0, rr = 0;
1145 static boolean scramble = false;
1146 private static void scramblePicker(boolean reset) {
1147     // rescrambles the pickers 50% of the time
1148     scramble = reset; // test scrambling it every time it is reset
1149     if(reset)
1150         scramble = Math.random() < 0.5;
1151 }
1152
1153 static boolean scrambleMethod = false;
1154 private static void scramblePickMethod(boolean reset) {
1155     // rescrambles the pickers 50% of the time
1156     if(reset)
1157         scrambleMethod = Math.random() < 0.5;
1158 }
1159 static boolean wildCard1 = false, wildCard2 = false;
1160 private static boolean pickMethod(boolean method1, boolean method2, boolean method3) {
1161     // takes a random pick of these 3 methods (and all possible combinations of
these)
1162     if(scrambleMethod) {
1163         r = Math.random();
1164         wildCard1 = r < 0.05;
1165         wildCard2 = r > 0.95;
1166     }
1167     if (wildCard1)
1168         return true; // very occasionally, pick any item at random to test with
a loaded price, in case it may be discovered to be a gem
1169     if(method1 == method2 == method3)
```

```
1170         return method1; // because it's Hobson's choice, so why waste processing
time deciding it?
1171         return picky(method1, method2, method3);
1172     }
1173
1174     private static boolean pickMethod2(boolean method1, boolean method2, boolean method3) {
1175         // keep the pick of the methods the same as the previous pickMethod(),
regardless of any possible scramble
1176         if (wildCard2)
1177             return true; // very occasionally, pick any item at random to test with
an unloaded price, in case it may be discovered to be a gem
1178         if(method1 == method2 == method3)
1179             return method1; // because it's Hobson's choice, so why waste processing
time deciding it?
1180         return picky(method1, method2, method3);
1181     }
1182
1183     private static boolean picky(boolean method1, boolean method2, boolean method3) {
1184         // to serve pickMethod() and pickMethod2()
1185         boolean[] combinations = {method1, method2, method3, method1 && method2,
method1 && method2 && method3, method1 && method3, method2 && method3};
1186         int i = 1;
1187         while(i <= combinations.length) {
1188             if(r < (double)i * 1.0 / combinations.length)
1189                 return combinations[i-1];
1190             i++;
1191         }
1192         return false; // should never reach here
1193     }
1194
1195     private static float pickrand(double...choices) {
1196         // returns a random pick of the choices provided
```

```

1197         if(scramble) // retains the same choice of the set provided, unless
scramblePicker() has been run and it has decided (with even odds) to rescrumble the
choice
1198             rr = Math.random();
1199             return pixie(choices);
1200     }
1201
1202     private static float pick(double...choices) {
1203         // returns a random pick of the choices provided, adopting the same pick of
this set as done with any previous pick
1204         return pixie(choices);
1205     }
1206
1207     static String R = "abcdefghijk";
1208     static char stick;
1209     private static float pixie(double[] choices) {
1210         // serves pickrand() and pick()
1211         int i = 1;
1212         while(i <= choices.length) {
1213             if(rr < (double)i * 1.0 / choices.length) {
1214                 stick = R.charAt(i-1);
1215                 return (float)choices[i-1];
1216             }
1217             i++;
1218         }
1219         return -1;
1220     }
1221
1222     private static double min(double[] array) {
1223         // Returns the smallest of a series of numbers
1224         double min = Double.MAX_VALUE;
1225         for(int i = 0; i < array.length; i++) {

```

```
1226         if(array[i] < min)
1227             min = array[i];
1228     }
1229     return min;
1230 }
1231
1232 private static double max(double[] array) {
1233     // Returns the largest of a series of numbers
1234     double max = Double.MIN_VALUE;
1235     for(int i = 0; i < array.length; i++) {
1236         if(array[i] > max)
1237             max = array[i];
1238     }
1239     return max;
1240 }
1241
1242 private static int count(float[] array) {
1243     // Returns the size of the array
1244     double total = 0;
1245     int i = array.length-1;
1246     while(i >= 0) {
1247         total += array[i];
1248         if(total > 0)
1249             return i+1;
1250         i--;
1251     }
1252     return 0;
1253 }
1254
1255 private static int count(double[] array) {
1256     // Returns the size of the array
1257     double total = 0;
```

```
1258     int i = array.length-1;
1259     while(i >= 0) {
1260         total += array[i];
1261         if(total > 0)
1262             return i+1;
1263         i--;
1264     }
1265     return 0;
1266 }
1267
1268 private static int count(double[][] array) {
1269     // Returns the number of non-zero elements in the array
1270     int count = 0;
1271     int i = array[0].length-1;
1272     while(i >= 0) {
1273         int j = array.length-1;
1274         while(j >= 0) {
1275             if(array[j][i] != 0)
1276                 count++;
1277             j--;
1278         }
1279         i--;
1280     }
1281     return count;
1282 }
1283
1284 private static float mean(float[][] array) {
1285     // Returns the mean of a series of numbers passed in a fully-populated 2D
array
1286     double total = 0;
1287     for(int i = 0; i < array.length; i++)
1288         for(int j = 0; j < array[0].length; j++)
```

```
1289         total += array[i][j];
1290     return (float)(total / (array.length * array[0].length));
1291 }
1292
1293 private static double mean(double[][] array) {
1294     // Returns the mean of a series of numbers passed in a fully-populated 2D
array
1295     double total = 0;
1296     for(int i = 0; i < array.length; i++)
1297         for(int j = 0; j < array[0].length; j++)
1298             total += array[i][j];
1299     return total / (array.length * array[0].length);
1300 }
1301
1302 private static float mean(float[] array) {
1303     // Returns the mean of a series of numbers passed in a sparsely-populated 1D
array
1304     double total = 0;
1305     int size = count(array);
1306     for(int i = 0; i < size; i++)
1307         total += array[i];
1308     return (float)(total / size);
1309 }
1310
1311 private static double mean(double[] array) {
1312     // Returns the mean of a series of numbers passed in a sparsely-populated 1D
array
1313     double total = 0;
1314     int size = count(array);
1315     for(int i = 0; i < size; i++)
1316         total += array[i];
1317     return total / size;
```

```
1318     }
1319
1320     private static double sqsum(float[] array) {
1321         // Returns the summation of the square value of a series of numbers
1322         double squareSum = 0;
1323         int size = count(array);
1324         for(int i = 0; i < size; i++)
1325             squareSum += Math.pow(array[i], 2);
1326         return squareSum;
1327     }
1328
1329     private static double sqsum(double[] array) {
1330         // Returns the summation of the square value of a series of numbers
1331         double squareSum = 0;
1332         int size = count(array);
1333         for(int i = 0; i < size; i++)
1334             squareSum += Math.pow(array[i], 2);
1335         return squareSum;
1336     }
1337
1338     private static double sqsum(double[][] array) {
1339         // Returns the summation of the square value of a series of numbers
1340         double squareSum = 0;
1341         for(int i = 0; i < array.length; i++)
1342             for(int j = 0; j < array[0].length; j++)
1343                 squareSum += Math.pow(array[i][j], 2);
1344         return squareSum;
1345     }
1346
1347     private static double sum(float[] array) {
1348         // Returns the summation of a series of numbers passed in an array
1349         double sum = 0;
```

```
1350     for(int i = 0; i < array.length; i++)
1351         sum += array[i];
1352     return sum;
1353 }
1354
1355 private static double sum(double[] array) {
1356     // Returns the summation of a series of numbers passed in an array
1357     double sum = 0;
1358     for(int i = 0; i < array.length; i++)
1359         sum += array[i];
1360     return sum;
1361 }
1362
1363 private static double sum(double[][] array) {
1364     // Returns the summation of a series of numbers passed in an array
1365     double sum = 0;
1366     for(int i = 0; i < array.length; i++)
1367         for(int j = 0; j < array[0].length; j++)
1368             sum += array[i][j];
1369     return sum;
1370 }
1371
1372 private static double scalar(float[] array) {
1373     return 1/(double)(count(array)-1);
1374 }
1375
1376 private static double scalar(double[] array) {
1377     return 1/(double)(count(array)-1);
1378 }
1379
1380 private static double scalar(double[][] array) {
1381     return 1/(double)(count(array)-1);
```

```
1382     }
1383
1384     private static double var(float[] array) {
1385         // Returns the variance of a series of numbers passed in an array
1386         return (scalar(array)*(sqsum(array) - (Math.pow(sum(array), 2)/count(array))));
1387     }
1388
1389     private static double var(double[] array) {
1390         // Returns the variance of a series of numbers passed in an array
1391         return (scalar(array)*(sqsum(array) - (Math.pow(sum(array), 2)/count(array))));
1392     }
1393
1394     private static double var(double[][] array) {
1395         // Returns the variance of a series of numbers passed in an array
1396         return (scalar(array)*(sqsum(array) - (Math.pow(sum(array), 2)/count(array))));
1397     }
1398
1399     private static double stdev(float[] array) {
1400         // Returns the standard deviation of the series of numbers passed in an
array
1401         return Math.sqrt(var(array));
1402     }
1403
1404     private static double stdev(double[] array) {
1405         // Returns the standard deviation of the series of numbers passed in an
array
1406         return Math.sqrt(var(array));
1407     }
1408
1409     private static double stdev(double[][] array) {
1410         // Returns the standard deviation of the series of numbers passed in an
array
```

```
1411     return Math.sqrt(var(array));
1412 }
1413
1414 private static float normrandHiLo(double lo, double hi) {
1415     // Returns a random value that is normally distributed
1416     double mean = (hi + lo) / 2.0;
1417     double stdev = (hi - lo) / 2.0;
1418     return (float)normrand(mean, stdev);
1419 }
1420
1421 private static double normrand(double mean, double stdev) {
1422     // Returns a random value that is normally distributed
1423     return mean + stdev*Math.sqrt(-
1424 1*Math.log(Math.random()))*Math.sin(2*Math.PI*Math.random());
1425 }
1426 private static double normsrand() {
1427     // Returns a standard random value that is normally distributed (that has a
1428 mean of zero and a standard deviation of one)
1429     return normrand(0.0, 1.0);
1430 }
1431 private static double normsdist(double z) {
1432     // Returns the standard normal cumulative distribution (has a mean of zero
1433 and a standard deviation of one)
1434     return normdist(z, 0.0F, 1.0F);
1435 }
1436 private static double normdist(double z, double mean, double spread) {
1437     double y;
1438     double stdev;
1439     double flatPortion = 0.5; // with this modification of the normal
```

```

distribution, we assume that 50% of the spread specified is "flat" - where there is no
risk whatsoever regardless of the price
1440     if(splitNORMALdistribution) {
1441         double flat = flatPortion * spread;
1442         stdev = spread - flat;
1443         if(z > mean - flat && z < mean + flat)
1444             y = (1.0/(stdev*Math.sqrt(2.0*Math.PI)));
1445         else {
1446             double zmean;
1447             if(z < mean)
1448                 zmean = z - mean + flat;
1449             else
1450                 zmean = z - mean - flat;
1451             y = (1.0/(stdev*Math.sqrt(2.0*Math.PI)))*Math.pow(10.0, (-
Math.pow(zmean,2)/(2.0*Math.pow(stdev,2)))); // complying to a normal distribution
1452         }
1453     } else {
1454         stdev = spread;
1455         y = (1.0/(stdev*Math.sqrt(2.0*Math.PI)))*Math.pow(10.0, (-Math.pow(z -
mean,2)/(2.0*Math.pow(stdev,2)))); // complying to a normal distribution
1456     }
1457     return y;
1458 }
1459
1460     private static double norminv(double probability, double mean, double stdev) {
1461         // Returns the inverse of the normal cumulative distribution for the
specified mean and standard deviation
1462         // See
https://secure.codeproject.com/script/Forums/View.aspx?fid=326859&msg=2624087
1463         double p = probability;
1464         if(probability >= 0.5)
1465             p = 1.0 - probability;

```

```

1466     double t = Math.sqrt(Math.log(1.0/Math.pow(p, 2)));
1467     double c_0 = 2.515517;
1468     double c_1 = 0.802853;
1469     double c_2 = 0.010328;
1470     double d_1 = 1.432788;
1471     double d_2 = 0.189269;
1472     double d_3 = 0.001308;
1473
1474     double X = t - (c_0 + c_1*t + c_2*Math.pow(t,2)) / (1.0 + d_1*t +
d_2*Math.pow(t,2) + d_3*Math.pow(t,3));
1475
1476     X = - X; // negate X
1477     if(probability >= 0.5)
1478         X = - X; // negate X
1479
1480     X *= stdev;
1481     X += mean;
1482
1483     return X; // Producing results roughly consistent with Excel
1484 }
1485
1486 private static void writeLine(FileOutputStream flog, String line) throws
java.io.IOException {
1487     byte b[] = line.getBytes();
1488     flog.write(b);
1489 }
1490
1491 private static void closeFlog(FileOutputStream flog) throws java.io.IOException {
1492     flushFlog(flog);
1493     flog.close();
1494 }
1495

```

```
1496     private static FileOutputStream openFlog(FileOutputStream flog, String name) throws
java.io.IOException {
1497         flushFlog(flog);
1498         File fileName = new File(name);
1499         fileName.delete();
1500         return new FileOutputStream(fileName, true);
1501     }
1502
1503     private static void flushFlog(FileOutputStream flog) throws java.io.IOException {
1504         FileDescriptor desc = flog.getFD();
1505         desc.sync(); // because flush only means that a high-level buffer gets
cleared but the data often can still sit in a system buffer without being written to the
actual disk surface. Sync'ing should ensure that the system buffers get flushed and the
data written to disk. Even when it's requested, it doesn't always happen without error
though.
1506         flog.flush();
1507     }
1508
1509     private static void numberOfItems(String fileName) {
1510         try {
1511             Scanner scanner = new Scanner(new File(fileName));
1512             scanner.useDelimiter(System.getProperty("line.separator"));
1513             int big = 0;
1514             int lineCounter = 0;
1515             while(scanner.hasNext()) {
1516                 big = index(scanner.next()+1;
1517                 int bigger = Math.max(big, lineCounter);
1518                 if(bigger > NUMBER_OF_ITEMS)
1519                     NUMBER_OF_ITEMS = bigger;
1520             }
1521             scanner.close();
1522         } catch(java.io.IOException ioe) {
```

```
1523         ioe.printStackTrace();
1524     }
1525 }
1526
1527 private static int index(String line) {
1528     Scanner lineScanner = new Scanner(line);
1529     lineScanner.useDelimiter("\\s*,\\s*");
1530     int index = lineScanner.nextInt();
1531     return index;
1532 }
1533
1534 private static float[][] readfile(String fileName) {
1535     final int COL = 8;
1536     float[][] array = new float[NUMBER_OF_ITEMS][COL+NUMBER_OF_MONTHS+1];
1537     float[] values = new float[COL+NUMBER_OF_MONTHS+1];
1538     try {
1539         Scanner scanner = new Scanner(new File(fileName));
1540         scanner.useDelimiter(System.getProperty("line.separator"));
1541         int index = 0;
1542         while(scanner.hasNext()) {
1543             values = parseLine(scanner.next(), index);
1544             for (int i = 0; i < COL+NUMBER_OF_MONTHS+1; i++)
1545                 array[index][i] = values[i];
1546             index++;
1547         }
1548         scanner.close();
1549     } catch (java.io.IOException ioe) {
1550         ioe.printStackTrace();
1551     }
1552     return array;
1553 }
1554
```

```
1555 private static float[] parseLine(String line, int in) {
1556     float[] values = new float[8+NUMBER_OF_MONTHS+1];
1557     String[] segments = new String[NUMBER_OF_MONTHS+1];
1558     if(line.endsWith("\r"))
1559         line = line.substring(0, line.length()-1);
1560     Scanner lineScanner = new Scanner(line);
1561     lineScanner.useDelimiter("\\s*,\\s*");
1562     values[0] = lineScanner.nextInt(); // index
1563     values[1] = lineScanner.nextInt(); // qnty
1564     values[2] = lineScanner.nextFloat(); // variance
1565     values[3] = lineScanner.nextFloat(); // varSD
1566     String wrkGrp = lineScanner.next();
1567     if(wrkGrp.equals("LOW"))
1568         values[4] = LOW;
1569     else if(wrkGrp.equals("HIGH"))
1570         values[4] = HIGH;
1571     else
1572         values[4] = MID;
1573     values[5] = lineScanner.nextFloat(); // cost
1574     values[6] = lineScanner.nextFloat(); // loPrice
1575     values[7] = lineScanner.nextFloat(); // hiPrice
1576     int i = 0;
1577     while(lineScanner.hasNext()) {
1578         segments[i++] = lineScanner.next();
1579     }
1580     boolean confident = segments[0].equals("true");
1581     if(confident)
1582         values[8] = 1;
1583     else
1584         values[8] = 0;
1585     int j = 1;
1586     while(j < i) {
```

```
1587     String[] sides = segments[j].split("=");
1588     float rhs = Float.parseFloat(sides[1]);
1589     String lhs = sides[0];
1590     if(lhs.indexOf("-") >= 0) {
1591         String[] mons = lhs.split("-");
1592         for(int m = Integer.parseInt(mons[0]); m <= Integer.parseInt(mons[1]);
m++)
1593             values[9+m] = rhs;
1594     } else
1595         values[9+Integer.parseInt(lhs)] = rhs;
1596     j++;
1597 }
1598 return values;
1599 }
1600 }
1601
1602
1603
```

APPENDIX B. HYPOTHETICAL PROJECT

INTRODUCTION

This appendix serves to describe the hypothetical project developed to serve as input data for the Xpload software. This project has been designed to serve as a test of the model proposed in this thesis. It includes a spectrum of artificial items created with a view to testing to examine how the model reacts to each of their circumstances. Typically, each item is designed to be similar to many other items and yet different in one regard. This facilitates the test to see how the model treats each of these items. The objective is, therefore, not only to see how the model handles the collective, overall project, but also how it prices each of the constituent items in the process.

There are 504 items (referenced as items 0–503) in the project and they are read by the software from a data file named “items.txt”.

This hypothetical project is not intended to resemble any particular real-world project nor is it intended to emulate any practical project that is likely to be encountered in the real world. It is, rather, specifically designed to serve to test robustness of the model to determine whether it performs as hypothesised.

ITEMS.TXT

This file contains the following fields of data for each of the 504 items that comprise the overall hypothetical project:

Item Number (e.g. 0)

A unique reference to each item, in the range 0 to 503.

Item Quantity in the Bills of Quantities (e.g. 100)

All items have been given the arbitrary quantity of 100. This represents the quantity prepared by the Quantity Surveyor as would be presented in the project's Bills of Quantities. These are often subject to variation during the course of the project.

Expected Variation (e.g. 1.3)

A value of 1.0 for this field would represent no variation from the Item Quantity whilst a value of 1.1, for example, would represent the volume of work increasing by 10%. Similarly, a value of 0.9 would represent a 10% decrease in the *quantum* of this item.

Standard Deviation in the estimate of this variation (e.g. 0.06)

This field gives the contractor an opportunity to indicate their confidence in their estimate of the aforementioned expected quantity variation. Three alternative scenarios are tested: 0.0 representing absolute confidence; 0.06 representing a moderate degree of confidence; and 0.15 representing a lesser-degree of confidence. The quantum of these degrees of confidence is not important and the purpose of this exercise is to test how the model will react to situations where there are simply differences, to some degree, in the extent to which the contractor has confidence in their estimate.

Nature of Workgroup (e.g. HIGH)

This field provides the opportunity to indicate whether the item in question is contained in a workgroup that has an expectation of high or low rates of escalation, with two possibilities of confidence in the estimates of these rates. A value of 'LOW' is used to describe an item with an expected rate of escalation of just 1% per annum (with a low

standard deviation of 0.1%); ‘MID’ is used to describe a workgroup with a mean expectation of a 12% rate of escalation, with a low rate of confidence reflecting in a relatively high 3.6% standard deviation); and ‘HIGH’ is used to describe a workgroup that also has a 12% rate of escalation where the confidence in this is, however, high (represented by a standard deviation of 1.2%). These values are hard-coded into the software.

As with the above-mentioned Standard Deviation, the quantum of these rates of escalation is not intended to be significant. The purpose of this hypothetical project is instead to test whether the model responds to differences in escalation rates between items in a manner that is heuristically consistent with the model’s underlying hypotheses.

Estimated Unit Cost (e.g. 110)

The model uses the estimated item unit costs for the purpose of determining the value of the ‘loss of reaction’ (see page 89 for an explanation).

Low Market-Price (e.g. 100)

The model requires that the contractor provide an indication of the likely reaction from the client to prices for this nature of item. Heuristics suggest that, for each item, there is a range of prices that are likely to be acceptable in the sense of not likely provoking any ‘reaction’ or ‘rejection’ from the client. For this purpose a client is defined as ‘reacting’ if the item price is such that it serves to induce them to redesign the project so as to either increase or decrease the quantity of this item. For instance, a client may adopt a view that an especially low price for an item represents an opportunity for them to redesign their project such that they take advantage of making increased use of this item. An example is if the price of one possible “finish” for a particular element is very low, the client may choose to switch the specification from alternative, more highly-priced finishes to this one.

Similarly, ‘rejection’ is defined as the situation where a client may be so outraged by the contractor’s pricing that they close negotiations with this contractor in favour of rather working with another contractor. The consequent ‘loss of rejection’ is therefore the cost

to the contractor of losing out on this opportunity. See pages 86-87 for an explanation of this opportunity cost.

This Low price is defined as the price-point at which there is roughly a 16% chance that a lesser price will be acceptable (by the above definition).

High Market-Price (e.g. 150)

Similarly, the model requires an indication of the upper-end range of acceptable pricing. If an item is too highly priced, this price could lead to a 'reaction' from the client. For instance, a client may adopt a view that an especially high price for an item represents an opportunity for them to redesign their project in order that they replace this item with another of a lesser price. If, for example, the price of one finish is considered especially high, the client may wish to go ahead with the project but change the specifications so that an alternative, lesser-priced finish is used instead.

A client could also 'reject' a high price and close negotiations with this contractor.

This High price is defined as the price-point at which there is roughly a 16% chance that a higher price will be acceptable (by the above definition).

Scheduling Confidence (e.g. true)

The next (boolean) field provides the contractor with the opportunity to provide indication of their view on whether they have a sense of confidence or not with regard to the scheduled timing for this item. This is used to assess the 'risk of being wrong'. If they are confident (i.e. their input is 'true') then the effect is for the software to assess this risk by way of applying a normal distribution with a standard deviation of 5% of the weighted-average 'central time' for that item. If they are instead not confident (i.e. the input is 'false') then it is assumed that the standard deviation is 20%, calculated on the same basis. This simple 2-way categorisation of all items is intended to test whether the model performs well in taking account of this nature of risk, in effect steering the pricing allocation away from risky items and rather placing higher prices on less risky items – when all other factors remain the same.

Schedule (e.g. 0-9=0.1)

The next combination of fields for each item serve to describe the scheduled timing of the item, entered as fractions of 1.0 where 1.0 represents 100% of the quantity of the item being built. For instance, if a contractor envisages building an item at such a time that 50% of it is complete in month 3 and 50% in month 4 then they could provide this as input in this file by way of the following segment of input:

3-4=0.5

Alternatively, the parser (written as part of the Xpload software) will also accept this input entered using the following syntax:

3=0.5, 4=0.5

The scheduling has predominantly been captured to indicate whether items are scheduled for early within a 30-month long schedule, in the middle of that project, or in the last 10 months of the project. Item price loading normally entails the “front-end loading” of items that are scheduled for early in the project, whilst “back-end loading” entails applying high prices to items that are expected to have a high escalation rate that are scheduled for late in the project. These processes are explained on pages 58-60 and 62-64, respectively.

Each item has data in each row for each of the above-mentioned fields, separated by commas.

The item data

0, 100, 1.3, 0.06, HIGH, 110, 100, 150, true, 0-9=0.1
 1, 100, 1.3, 0.06, HIGH, 100, 130, 140, true, 0-9=0.1
 2, 100, 1.3, 0.06, HIGH, 110, 100, 150, true, 0-9=0.1
 3, 100, 1.3, 0.06, HIGH, 110, 100, 110, true, 0-9=0.1
 4, 100, 1.3, 0.15, HIGH, 100, 100, 150, true, 0-9=0.1
 5, 100, 1.3, 0.15, HIGH, 100, 130, 140, true, 0-9=0.1
 6, 100, 1.3, 0.15, HIGH, 105, 100, 150, true, 0-9=0.1
 7, 100, 1.3, 0.15, HIGH, 105, 100, 110, true, 0-9=0.1
 8, 100, 1.3, 0.06, HIGH, 110, 100, 150, true, 1=1.0
 9, 100, 1.3, 0.06, HIGH, 100, 130, 140, true, 1=1.0
 10, 100, 1.3, 0.06, HIGH, 110, 100, 150, true, 1=1.0
 11, 100, 1.3, 0.06, HIGH, 105, 100, 110, true, 1=1.0
 12, 100, 1.0, 0.0, HIGH, 110, 100, 150, true, 0-9=0.1
 13, 100, 1.0, 0.0, HIGH, 110, 130, 140, true, 0-9=0.1
 14, 100, 1.0, 0.0, HIGH, 120, 100, 150, true, 0-9=0.1
 15, 100, 1.0, 0.0, HIGH, 100, 100, 110, true, 0-9=0.1
 16, 100, 0.7, 0.06, HIGH, 120, 100, 150, true, 0-9=0.1
 17, 100, 0.7, 0.06, HIGH, 120, 130, 140, true, 0-9=0.1
 18, 100, 0.7, 0.06, HIGH, 120, 100, 150, true, 0-9=0.1
 19, 100, 0.7, 0.06, HIGH, 100, 100, 110, true, 0-9=0.1
 20, 100, 0.7, 0.15, HIGH, 120, 100, 150, true, 0-9=0.1
 21, 100, 0.7, 0.15, HIGH, 120, 130, 140, true, 0-9=0.1
 22, 100, 0.7, 0.15, HIGH, 110, 100, 150, true, 0-9=0.1
 23, 100, 0.7, 0.15, HIGH, 100, 100, 110, true, 0-9=0.1
 24, 100, 0.7, 0.0, HIGH, 120, 100, 150, true, 0-9=0.1
 25, 100, 0.7, 0.0, HIGH, 120, 130, 140, true, 0-9=0.1
 26, 100, 0.7, 0.0, HIGH, 120, 100, 150, true, 0-9=0.1
 27, 100, 0.7, 0.0, HIGH, 100, 100, 110, true, 0-9=0.1
 28, 100, 1.3, 0.06, MID, 120, 100, 150, true, 0-9=0.1
 29, 100, 1.3, 0.06, MID, 120, 130, 140, true, 0-9=0.1
 30, 100, 1.3, 0.06, MID, 120, 100, 150, true, 0-9=0.1
 31, 100, 1.3, 0.06, MID, 100, 100, 110, true, 0-9=0.1
 32, 100, 1.3, 0.15, MID, 120, 100, 150, true, 0-9=0.1
 33, 100, 1.3, 0.15, MID, 120, 130, 140, true, 0-9=0.1
 34, 100, 1.3, 0.15, MID, 120, 100, 150, true, 0-9=0.1
 35, 100, 1.3, 0.15, MID, 100, 100, 110, true, 0-9=0.1
 36, 100, 1.3, 0.06, MID, 120, 100, 150, true, 0-9=0.1
 37, 100, 1.3, 0.06, MID, 120, 130, 140, true, 0-9=0.1
 38, 100, 1.3, 0.06, MID, 120, 100, 150, true, 0-9=0.1
 39, 100, 1.3, 0.06, MID, 100, 100, 110, true, 0-9=0.1
 40, 100, 1.0, 0.0, MID, 120, 100, 150, true, 0-9=0.1
 41, 100, 1.0, 0.0, MID, 120, 130, 140, true, 0-9=0.1
 42, 100, 1.0, 0.0, MID, 120, 100, 150, true, 0-9=0.1
 43, 100, 1.0, 0.0, MID, 100, 100, 110, true, 0-9=0.1
 44, 100, 0.7, 0.06, MID, 120, 100, 150, true, 0-9=0.1
 45, 100, 0.7, 0.06, MID, 120, 130, 140, true, 0-9=0.1

46, 100, 0.7, 0.06, MID, 120, 100, 150, true, 0-9=0.1
47, 100, 0.7, 0.06, MID, 100, 100, 110, true, 0-9=0.1
48, 100, 0.7, 0.15, MID, 120, 100, 150, true, 0-9=0.1
49, 100, 0.7, 0.15, MID, 120, 130, 140, true, 0-9=0.1
50, 100, 0.7, 0.15, MID, 120, 100, 150, true, 0-9=0.1
51, 100, 0.7, 0.15, MID, 100, 100, 110, true, 0-9=0.1
52, 100, 0.7, 0.0, MID, 120, 100, 150, true, 0-9=0.1
53, 100, 0.7, 0.0, MID, 120, 130, 140, true, 0-9=0.1
54, 100, 0.7, 0.0, MID, 120, 100, 150, true, 0-9=0.1
55, 100, 0.7, 0.0, MID, 100, 100, 110, true, 0-9=0.1
56, 100, 1.3, 0.06, LOW, 120, 100, 150, true, 0-9=0.1
57, 100, 1.3, 0.06, LOW, 120, 130, 140, true, 0-9=0.1
58, 100, 1.3, 0.06, LOW, 120, 100, 150, true, 0-9=0.1
59, 100, 1.3, 0.06, LOW, 100, 100, 110, true, 0-9=0.1
60, 100, 1.3, 0.15, LOW, 120, 100, 150, true, 0-9=0.1
61, 100, 1.3, 0.15, LOW, 120, 130, 140, true, 0-9=0.1
62, 100, 1.3, 0.15, LOW, 120, 100, 150, true, 0-9=0.1
63, 100, 1.3, 0.15, LOW, 100, 100, 110, true, 0-9=0.1
64, 100, 1.3, 0.06, LOW, 120, 100, 150, true, 0-9=0.1
65, 100, 1.3, 0.06, LOW, 120, 130, 140, true, 0-9=0.1
66, 100, 1.3, 0.06, LOW, 120, 100, 150, true, 0-9=0.1
67, 100, 1.3, 0.06, LOW, 100, 100, 110, true, 0-9=0.1
68, 100, 1.0, 0.0, LOW, 120, 100, 150, true, 0-9=0.1
69, 100, 1.0, 0.0, LOW, 120, 130, 140, true, 0-9=0.1
70, 100, 1.0, 0.0, LOW, 120, 100, 150, true, 0-9=0.1
71, 100, 1.0, 0.0, LOW, 100, 100, 110, true, 0-9=0.1
72, 100, 0.7, 0.06, LOW, 120, 100, 150, true, 0-9=0.1
73, 100, 0.7, 0.06, LOW, 120, 130, 140, true, 0-9=0.1
74, 100, 0.7, 0.06, LOW, 120, 100, 150, true, 0-9=0.1
75, 100, 0.7, 0.06, LOW, 100, 100, 110, true, 0-9=0.1
76, 100, 0.7, 0.15, LOW, 120, 100, 150, true, 0-9=0.1
77, 100, 0.7, 0.15, LOW, 120, 130, 140, true, 0-9=0.1
78, 100, 0.7, 0.15, LOW, 120, 100, 150, true, 0-9=0.1
79, 100, 0.7, 0.15, LOW, 100, 100, 110, true, 0-9=0.1
80, 100, 0.7, 0.0, LOW, 120, 100, 150, true, 0-9=0.1
81, 100, 0.7, 0.0, LOW, 120, 130, 140, true, 0-9=0.1
82, 100, 0.7, 0.0, LOW, 120, 100, 150, true, 0-9=0.1
83, 100, 0.7, 0.0, LOW, 100, 100, 110, true, 0-9=0.1
84, 100, 1.3, 0.06, HIGH, 120, 100, 150, true, 10-19=0.1
85, 100, 1.3, 0.06, HIGH, 120, 130, 140, true, 10-19=0.1
86, 100, 1.3, 0.06, HIGH, 120, 100, 150, true, 10-19=0.1
87, 100, 1.3, 0.06, HIGH, 100, 100, 110, true, 10-19=0.1
88, 100, 1.3, 0.15, HIGH, 120, 100, 150, true, 10-19=0.1
89, 100, 1.3, 0.15, HIGH, 120, 130, 140, true, 10-19=0.1
90, 100, 1.3, 0.15, HIGH, 120, 100, 150, true, 10-19=0.1
91, 100, 1.3, 0.15, HIGH, 100, 100, 110, true, 10-19=0.1
92, 100, 1.3, 0.06, HIGH, 120, 100, 150, true, 15=1.0
93, 100, 1.3, 0.06, HIGH, 120, 130, 140, true, 15=1.0

94, 100, 1.3, 0.06, HIGH, 120, 100, 150, true, 15=1.0
95, 100, 1.3, 0.06, HIGH, 100, 100, 110, true, 15=1.0
96, 100, 1.0, 0.0, HIGH, 120, 100, 150, true, 10-19=0.1
97, 100, 1.0, 0.0, HIGH, 120, 130, 140, true, 10-19=0.1
98, 100, 1.0, 0.0, HIGH, 120, 100, 150, true, 10-19=0.1
99, 100, 1.0, 0.0, HIGH, 100, 100, 110, true, 10-19=0.1
100, 100, 0.7, 0.06, HIGH, 120, 100, 150, true, 10-19=0.1
101, 100, 0.7, 0.06, HIGH, 120, 130, 140, true, 10-19=0.1
102, 100, 0.7, 0.06, HIGH, 120, 100, 150, true, 10-19=0.1
103, 100, 0.7, 0.06, HIGH, 100, 100, 110, true, 10-19=0.1
104, 100, 0.7, 0.15, HIGH, 120, 100, 150, true, 10-19=0.1
105, 100, 0.7, 0.15, HIGH, 120, 130, 140, true, 10-19=0.1
106, 100, 0.7, 0.15, HIGH, 120, 100, 150, true, 10-19=0.1
107, 100, 0.7, 0.15, HIGH, 100, 100, 110, true, 10-19=0.1
108, 100, 0.7, 0.0, HIGH, 120, 100, 150, true, 10-19=0.1
109, 100, 0.7, 0.0, HIGH, 120, 130, 140, true, 10-19=0.1
110, 100, 0.7, 0.0, HIGH, 120, 100, 150, true, 10-19=0.1
111, 100, 0.7, 0.0, HIGH, 100, 100, 110, true, 10-19=0.1
112, 100, 1.3, 0.06, MID, 120, 100, 150, true, 10-19=0.1
113, 100, 1.3, 0.06, MID, 120, 130, 140, true, 10-19=0.1
114, 100, 1.3, 0.06, MID, 120, 100, 150, true, 10-19=0.1
115, 100, 1.3, 0.06, MID, 100, 100, 110, true, 10-19=0.1
116, 100, 1.3, 0.15, MID, 120, 100, 150, true, 10-19=0.1
117, 100, 1.3, 0.15, MID, 120, 130, 140, true, 10-19=0.1
118, 100, 1.3, 0.15, MID, 120, 100, 150, true, 10-19=0.1
119, 100, 1.3, 0.15, MID, 100, 100, 110, true, 10-19=0.1
120, 100, 1.3, 0.06, MID, 120, 100, 150, true, 10-19=0.1
121, 100, 1.3, 0.06, MID, 120, 130, 140, true, 10-19=0.1
122, 100, 1.3, 0.06, MID, 120, 100, 150, true, 10-19=0.1
123, 100, 1.3, 0.06, MID, 100, 100, 110, true, 10-19=0.1
124, 100, 1.0, 0.0, MID, 120, 100, 150, true, 10-19=0.1
125, 100, 1.0, 0.0, MID, 120, 130, 140, true, 10-19=0.1
126, 100, 1.0, 0.0, MID, 120, 100, 150, true, 10-19=0.1
127, 100, 1.0, 0.0, MID, 100, 100, 110, true, 10-19=0.1
128, 100, 0.7, 0.06, MID, 120, 100, 150, true, 10-19=0.1
129, 100, 0.7, 0.06, MID, 120, 130, 140, true, 10-19=0.1
130, 100, 0.7, 0.06, MID, 120, 100, 150, true, 10-19=0.1
131, 100, 0.7, 0.06, MID, 100, 100, 110, true, 10-19=0.1
132, 100, 0.7, 0.15, MID, 120, 100, 150, true, 10-19=0.1
133, 100, 0.7, 0.15, MID, 120, 130, 140, true, 10-19=0.1
134, 100, 0.7, 0.15, MID, 120, 100, 150, true, 10-19=0.1
135, 100, 0.7, 0.15, MID, 100, 100, 110, true, 10-19=0.1
136, 100, 0.7, 0.0, MID, 120, 100, 150, true, 10-19=0.1
137, 100, 0.7, 0.0, MID, 120, 130, 140, true, 10-19=0.1
138, 100, 0.7, 0.0, MID, 120, 100, 150, true, 10-19=0.1
139, 100, 0.7, 0.0, MID, 100, 100, 110, true, 10-19=0.1
140, 100, 1.3, 0.06, LOW, 120, 100, 150, true, 10-19=0.1
141, 100, 1.3, 0.06, LOW, 120, 130, 140, true, 10-19=0.1

142, 100, 1.3, 0.06, LOW, 120, 100, 150, true, 10-19=0.1
143, 100, 1.3, 0.06, LOW, 100, 100, 110, true, 10-19=0.1
144, 100, 1.3, 0.15, LOW, 120, 100, 150, true, 10-19=0.1
145, 100, 1.3, 0.15, LOW, 120, 130, 140, true, 10-19=0.1
146, 100, 1.3, 0.15, LOW, 120, 100, 150, true, 10-19=0.1
147, 100, 1.3, 0.15, LOW, 100, 100, 110, true, 10-19=0.1
148, 100, 1.3, 0.06, LOW, 120, 100, 150, true, 10-19=0.1
149, 100, 1.3, 0.06, LOW, 120, 130, 140, true, 10-19=0.1
150, 100, 1.3, 0.06, LOW, 120, 100, 150, true, 10-19=0.1
151, 100, 1.3, 0.06, LOW, 100, 100, 110, true, 10-19=0.1
152, 100, 1.0, 0.0, LOW, 120, 100, 150, true, 10-19=0.1
153, 100, 1.0, 0.0, LOW, 120, 130, 140, true, 10-19=0.1
154, 100, 1.0, 0.0, LOW, 120, 100, 150, true, 10-19=0.1
155, 100, 1.0, 0.0, LOW, 100, 100, 110, true, 10-19=0.1
156, 100, 0.7, 0.06, LOW, 120, 100, 150, true, 10-19=0.1
157, 100, 0.7, 0.06, LOW, 120, 130, 140, true, 10-19=0.1
158, 100, 0.7, 0.06, LOW, 120, 100, 150, true, 10-19=0.1
159, 100, 0.7, 0.06, LOW, 100, 100, 110, true, 10-19=0.1
160, 100, 0.7, 0.15, LOW, 120, 100, 150, true, 10-19=0.1
161, 100, 0.7, 0.15, LOW, 120, 130, 140, true, 10-19=0.1
162, 100, 0.7, 0.15, LOW, 120, 100, 150, true, 10-19=0.1
163, 100, 0.7, 0.15, LOW, 100, 100, 110, true, 10-19=0.1
164, 100, 0.7, 0.0, LOW, 120, 100, 150, true, 10-19=0.1
165, 100, 0.7, 0.0, LOW, 120, 130, 140, true, 10-19=0.1
166, 100, 0.7, 0.0, LOW, 120, 100, 150, true, 10-19=0.1
167, 100, 0.7, 0.0, LOW, 100, 100, 110, true, 10-19=0.1
168, 100, 1.3, 0.06, HIGH, 120, 100, 150, true, 20-29=0.1
169, 100, 1.3, 0.06, HIGH, 120, 130, 140, true, 20-29=0.1
170, 100, 1.3, 0.06, HIGH, 120, 100, 150, true, 20-29=0.1
171, 100, 1.3, 0.06, HIGH, 100, 100, 110, true, 20-29=0.1
172, 100, 1.3, 0.15, HIGH, 120, 100, 150, true, 20-29=0.1
173, 100, 1.3, 0.15, HIGH, 120, 130, 140, true, 20-29=0.1
174, 100, 1.3, 0.15, HIGH, 120, 100, 150, true, 20-29=0.1
175, 100, 1.3, 0.15, HIGH, 100, 100, 110, true, 20-29=0.1
176, 100, 1.3, 0.06, HIGH, 120, 100, 150, true, 29=1.0
177, 100, 1.3, 0.06, HIGH, 120, 130, 140, true, 29=1.0
178, 100, 1.3, 0.06, HIGH, 120, 100, 150, true, 29=1.0
179, 100, 1.3, 0.06, HIGH, 100, 100, 110, true, 29=1.0
180, 100, 1.0, 0.0, HIGH, 120, 100, 150, true, 20-29=0.1
181, 100, 1.0, 0.0, HIGH, 120, 130, 140, true, 20-29=0.1
182, 100, 1.0, 0.0, HIGH, 120, 100, 150, true, 20-29=0.1
183, 100, 1.0, 0.0, HIGH, 100, 100, 110, true, 20-29=0.1
184, 100, 0.7, 0.06, HIGH, 120, 100, 150, true, 20-29=0.1
185, 100, 0.7, 0.06, HIGH, 120, 130, 140, true, 20-29=0.1
186, 100, 0.7, 0.06, HIGH, 120, 100, 150, true, 20-29=0.1
187, 100, 0.7, 0.06, HIGH, 100, 100, 110, true, 20-29=0.1
188, 100, 0.7, 0.15, HIGH, 120, 100, 150, true, 20-29=0.1
189, 100, 0.7, 0.15, HIGH, 120, 130, 140, true, 20-29=0.1

190, 100, 0.7, 0.15, HIGH, 120, 100, 150, true, 20-29=0.1
191, 100, 0.7, 0.15, HIGH, 100, 100, 110, true, 20-29=0.1
192, 100, 0.7, 0.0, HIGH, 120, 100, 150, true, 20-29=0.1
193, 100, 0.7, 0.0, HIGH, 120, 130, 140, true, 20-29=0.1
194, 100, 0.7, 0.0, HIGH, 120, 100, 150, true, 20-29=0.1
195, 100, 0.7, 0.0, HIGH, 100, 100, 110, true, 20-29=0.1
196, 100, 1.3, 0.06, MID, 120, 100, 150, true, 20-29=0.1
197, 100, 1.3, 0.06, MID, 120, 130, 140, true, 20-29=0.1
198, 100, 1.3, 0.06, MID, 120, 100, 150, true, 20-29=0.1
199, 100, 1.3, 0.06, MID, 100, 100, 110, true, 20-29=0.1
200, 100, 1.3, 0.15, MID, 120, 100, 150, true, 20-29=0.1
201, 100, 1.3, 0.15, MID, 120, 130, 140, true, 20-29=0.1
202, 100, 1.3, 0.15, MID, 120, 100, 150, true, 20-29=0.1
203, 100, 1.3, 0.15, MID, 100, 100, 110, true, 20-29=0.1
204, 100, 1.3, 0.06, MID, 120, 100, 150, true, 20-29=0.1
205, 100, 1.3, 0.06, MID, 120, 130, 140, true, 20-29=0.1
206, 100, 1.3, 0.06, MID, 120, 100, 150, true, 20-29=0.1
207, 100, 1.3, 0.06, MID, 100, 100, 110, true, 20-29=0.1
208, 100, 1.0, 0.0, MID, 120, 100, 150, true, 20-29=0.1
209, 100, 1.0, 0.0, MID, 120, 130, 140, true, 20-29=0.1
210, 100, 1.0, 0.0, MID, 120, 100, 150, true, 20-29=0.1
211, 100, 1.0, 0.0, MID, 100, 100, 110, true, 20-29=0.1
212, 100, 0.7, 0.06, MID, 120, 100, 150, true, 20-29=0.1
213, 100, 0.7, 0.06, MID, 120, 130, 140, true, 20-29=0.1
214, 100, 0.7, 0.06, MID, 120, 100, 150, true, 20-29=0.1
215, 100, 0.7, 0.06, MID, 100, 100, 110, true, 20-29=0.1
216, 100, 0.7, 0.15, MID, 120, 100, 150, true, 20-29=0.1
217, 100, 0.7, 0.15, MID, 120, 130, 140, true, 20-29=0.1
218, 100, 0.7, 0.15, MID, 120, 100, 150, true, 20-29=0.1
219, 100, 0.7, 0.15, MID, 100, 100, 110, true, 20-29=0.1
220, 100, 0.7, 0.0, MID, 120, 100, 150, true, 20-29=0.1
221, 100, 0.7, 0.0, MID, 120, 130, 140, true, 20-29=0.1
222, 100, 0.7, 0.0, MID, 120, 100, 150, true, 20-29=0.1
223, 100, 0.7, 0.0, MID, 100, 100, 110, true, 20-29=0.1
224, 100, 1.3, 0.06, LOW, 120, 100, 150, true, 20-29=0.1
225, 100, 1.3, 0.06, LOW, 120, 130, 140, true, 20-29=0.1
226, 100, 1.3, 0.06, LOW, 120, 100, 150, true, 20-29=0.1
227, 100, 1.3, 0.06, LOW, 100, 100, 110, true, 20-29=0.1
228, 100, 1.3, 0.15, LOW, 120, 100, 150, true, 20-29=0.1
229, 100, 1.3, 0.15, LOW, 120, 130, 140, true, 20-29=0.1
230, 100, 1.3, 0.15, LOW, 120, 100, 150, true, 20-29=0.1
231, 100, 1.3, 0.15, LOW, 100, 100, 110, true, 20-29=0.1
232, 100, 1.3, 0.06, LOW, 120, 100, 150, true, 20-29=0.1
233, 100, 1.3, 0.06, LOW, 120, 130, 140, true, 20-29=0.1
234, 100, 1.3, 0.06, LOW, 120, 100, 150, true, 20-29=0.1
235, 100, 1.3, 0.06, LOW, 100, 100, 110, true, 20-29=0.1
236, 100, 1.0, 0.0, LOW, 120, 100, 150, true, 20-29=0.1
237, 100, 1.0, 0.0, LOW, 120, 130, 140, true, 20-29=0.1

238, 100, 1.0, 0.0, LOW, 120, 100, 150, true, 20-29=0.1
239, 100, 1.0, 0.0, LOW, 100, 100, 110, true, 20-29=0.1
240, 100, 0.7, 0.06, LOW, 120, 100, 150, true, 20-29=0.1
241, 100, 0.7, 0.06, LOW, 120, 130, 140, true, 20-29=0.1
242, 100, 0.7, 0.06, LOW, 120, 100, 150, true, 20-29=0.1
243, 100, 0.7, 0.06, LOW, 100, 100, 110, true, 20-29=0.1
244, 100, 0.7, 0.15, LOW, 120, 100, 150, true, 20-29=0.1
245, 100, 0.7, 0.15, LOW, 120, 130, 140, true, 20-29=0.1
246, 100, 0.7, 0.15, LOW, 120, 100, 150, true, 20-29=0.1
247, 100, 0.7, 0.15, LOW, 100, 100, 110, true, 20-29=0.1
248, 100, 0.7, 0.0, LOW, 120, 100, 150, true, 20-29=0.1
249, 100, 0.7, 0.0, LOW, 120, 130, 140, true, 20-29=0.1
250, 100, 0.7, 0.0, LOW, 120, 100, 150, true, 20-29=0.1
251, 100, 0.7, 0.0, LOW, 100, 100, 110, true, 20-29=0.1
252, 100, 1.3, 0.06, HIGH, 120, 100, 150, false, 0-9=0.1
253, 100, 1.3, 0.06, HIGH, 120, 130, 140, false, 0-9=0.1
254, 100, 1.3, 0.06, HIGH, 120, 100, 150, false, 0-9=0.1
255, 100, 1.3, 0.06, HIGH, 100, 100, 110, false, 0-9=0.1
256, 100, 1.3, 0.15, HIGH, 120, 100, 150, false, 0-9=0.1
257, 100, 1.3, 0.15, HIGH, 120, 130, 140, false, 0-9=0.1
258, 100, 1.3, 0.15, HIGH, 120, 100, 150, false, 0-9=0.1
259, 100, 1.3, 0.15, HIGH, 100, 100, 110, false, 0-9=0.1
260, 100, 1.3, 0.06, HIGH, 120, 100, 150, false, 1=1.0
261, 100, 1.3, 0.06, HIGH, 120, 130, 140, false, 1=1.0
262, 100, 1.3, 0.06, HIGH, 120, 100, 150, false, 1=1.0
263, 100, 1.3, 0.06, HIGH, 100, 100, 110, false, 1=1.0
264, 100, 1.0, 0.0, HIGH, 120, 100, 150, false, 0-9=0.1
265, 100, 1.0, 0.0, HIGH, 120, 130, 140, false, 0-9=0.1
266, 100, 1.0, 0.0, HIGH, 120, 100, 150, false, 0-9=0.1
267, 100, 1.0, 0.0, HIGH, 100, 100, 110, false, 0-9=0.1
268, 100, 0.7, 0.06, HIGH, 120, 100, 150, false, 0-9=0.1
269, 100, 0.7, 0.06, HIGH, 120, 130, 140, false, 0-9=0.1
270, 100, 0.7, 0.06, HIGH, 120, 100, 150, false, 0-9=0.1
271, 100, 0.7, 0.06, HIGH, 100, 100, 110, false, 0-9=0.1
272, 100, 0.7, 0.15, HIGH, 120, 100, 150, false, 0-9=0.1
273, 100, 0.7, 0.15, HIGH, 120, 130, 140, false, 0-9=0.1
274, 100, 0.7, 0.15, HIGH, 120, 100, 150, false, 0-9=0.1
275, 100, 0.7, 0.15, HIGH, 100, 100, 110, false, 0-9=0.1
276, 100, 0.7, 0.0, HIGH, 120, 100, 150, false, 0-9=0.1
277, 100, 0.7, 0.0, HIGH, 120, 130, 140, false, 0-9=0.1
278, 100, 0.7, 0.0, HIGH, 120, 100, 150, false, 0-9=0.1
279, 100, 0.7, 0.0, HIGH, 100, 100, 110, false, 0-9=0.1
280, 100, 1.3, 0.06, MID, 120, 100, 150, false, 0-9=0.1
281, 100, 1.3, 0.06, MID, 120, 130, 140, false, 0-9=0.1
282, 100, 1.3, 0.06, MID, 120, 100, 150, false, 0-9=0.1
283, 100, 1.3, 0.06, MID, 100, 100, 110, false, 0-9=0.1
284, 100, 1.3, 0.15, MID, 120, 100, 150, false, 0-9=0.1
285, 100, 1.3, 0.15, MID, 120, 130, 140, false, 0-9=0.1

286, 100, 1.3, 0.15, MID, 120, 100, 150, false, 0-9=0.1
287, 100, 1.3, 0.15, MID, 100, 100, 110, false, 0-9=0.1
288, 100, 1.3, 0.06, MID, 120, 100, 150, false, 0-9=0.1
289, 100, 1.3, 0.06, MID, 120, 130, 140, false, 0-9=0.1
290, 100, 1.3, 0.06, MID, 120, 100, 150, false, 0-9=0.1
291, 100, 1.3, 0.06, MID, 100, 100, 110, false, 0-9=0.1
292, 100, 1.0, 0.0, MID, 120, 100, 150, false, 0-9=0.1
293, 100, 1.0, 0.0, MID, 120, 130, 140, false, 0-9=0.1
294, 100, 1.0, 0.0, MID, 120, 100, 150, false, 0-9=0.1
295, 100, 1.0, 0.0, MID, 100, 100, 110, false, 0-9=0.1
296, 100, 0.7, 0.06, MID, 120, 100, 150, false, 0-9=0.1
297, 100, 0.7, 0.06, MID, 120, 130, 140, false, 0-9=0.1
298, 100, 0.7, 0.06, MID, 120, 100, 150, false, 0-9=0.1
299, 100, 0.7, 0.06, MID, 100, 100, 110, false, 0-9=0.1
300, 100, 0.7, 0.15, MID, 120, 100, 150, false, 0-9=0.1
301, 100, 0.7, 0.15, MID, 120, 130, 140, false, 0-9=0.1
302, 100, 0.7, 0.15, MID, 120, 100, 150, false, 0-9=0.1
303, 100, 0.7, 0.15, MID, 100, 100, 110, false, 0-9=0.1
304, 100, 0.7, 0.0, MID, 120, 100, 150, false, 0-9=0.1
305, 100, 0.7, 0.0, MID, 120, 130, 140, false, 0-9=0.1
306, 100, 0.7, 0.0, MID, 120, 100, 150, false, 0-9=0.1
307, 100, 0.7, 0.0, MID, 100, 100, 110, false, 0-9=0.1
308, 100, 1.3, 0.06, LOW, 120, 100, 150, false, 0-9=0.1
309, 100, 1.3, 0.06, LOW, 120, 130, 140, false, 0-9=0.1
310, 100, 1.3, 0.06, LOW, 120, 100, 150, false, 0-9=0.1
311, 100, 1.3, 0.06, LOW, 100, 100, 110, false, 0-9=0.1
312, 100, 1.3, 0.15, LOW, 120, 100, 150, false, 0-9=0.1
313, 100, 1.3, 0.15, LOW, 120, 130, 140, false, 0-9=0.1
314, 100, 1.3, 0.15, LOW, 120, 100, 150, false, 0-9=0.1
315, 100, 1.3, 0.15, LOW, 100, 100, 110, false, 0-9=0.1
316, 100, 1.3, 0.06, LOW, 120, 100, 150, false, 0-9=0.1
317, 100, 1.3, 0.06, LOW, 120, 130, 140, false, 0-9=0.1
318, 100, 1.3, 0.06, LOW, 120, 100, 150, false, 0-9=0.1
319, 100, 1.3, 0.06, LOW, 100, 100, 110, false, 0-9=0.1
320, 100, 1.0, 0.0, LOW, 120, 100, 150, false, 0-9=0.1
321, 100, 1.0, 0.0, LOW, 120, 130, 140, false, 0-9=0.1
322, 100, 1.0, 0.0, LOW, 120, 100, 150, false, 0-9=0.1
323, 100, 1.0, 0.0, LOW, 100, 100, 110, false, 0-9=0.1
324, 100, 0.7, 0.06, LOW, 120, 100, 150, false, 0-9=0.1
325, 100, 0.7, 0.06, LOW, 120, 130, 140, false, 0-9=0.1
326, 100, 0.7, 0.06, LOW, 120, 100, 150, false, 0-9=0.1
327, 100, 0.7, 0.06, LOW, 100, 100, 110, false, 0-9=0.1
328, 100, 0.7, 0.15, LOW, 120, 100, 150, false, 0-9=0.1
329, 100, 0.7, 0.15, LOW, 120, 130, 140, false, 0-9=0.1
330, 100, 0.7, 0.15, LOW, 120, 100, 150, false, 0-9=0.1
331, 100, 0.7, 0.15, LOW, 100, 100, 110, false, 0-9=0.1
332, 100, 0.7, 0.0, LOW, 120, 100, 150, false, 0-9=0.1
333, 100, 0.7, 0.0, LOW, 120, 130, 140, false, 0-9=0.1

334, 100, 0.7, 0.0, LOW, 120, 100, 150, false, 0-9=0.1
335, 100, 0.7, 0.0, LOW, 100, 100, 110, false, 0-9=0.1
336, 100, 1.3, 0.06, HIGH, 120, 100, 150, false, 10-19=0.1
337, 100, 1.3, 0.06, HIGH, 120, 130, 140, false, 10-19=0.1
338, 100, 1.3, 0.06, HIGH, 120, 100, 150, false, 10-19=0.1
339, 100, 1.3, 0.06, HIGH, 100, 100, 110, false, 10-19=0.1
340, 100, 1.3, 0.15, HIGH, 120, 100, 150, false, 10-19=0.1
341, 100, 1.3, 0.15, HIGH, 120, 130, 140, false, 10-19=0.1
342, 100, 1.3, 0.15, HIGH, 120, 100, 150, false, 10-19=0.1
343, 100, 1.3, 0.15, HIGH, 100, 100, 110, false, 10-19=0.1
344, 100, 1.3, 0.06, HIGH, 120, 100, 150, false, 15=1.0
345, 100, 1.3, 0.06, HIGH, 120, 130, 140, false, 15=1.0
346, 100, 1.3, 0.06, HIGH, 120, 100, 150, false, 15=1.0
347, 100, 1.3, 0.06, HIGH, 100, 100, 110, false, 15=1.0
348, 100, 1.0, 0.0, HIGH, 120, 100, 150, false, 10-19=0.1
349, 100, 1.0, 0.0, HIGH, 120, 130, 140, false, 10-19=0.1
350, 100, 1.0, 0.0, HIGH, 120, 100, 150, false, 10-19=0.1
351, 100, 1.0, 0.0, HIGH, 100, 100, 110, false, 10-19=0.1
352, 100, 0.7, 0.06, HIGH, 120, 100, 150, false, 10-19=0.1
353, 100, 0.7, 0.06, HIGH, 120, 130, 140, false, 10-19=0.1
354, 100, 0.7, 0.06, HIGH, 120, 100, 150, false, 10-19=0.1
355, 100, 0.7, 0.06, HIGH, 100, 100, 110, false, 10-19=0.1
356, 100, 0.7, 0.15, HIGH, 120, 100, 150, false, 10-19=0.1
357, 100, 0.7, 0.15, HIGH, 120, 130, 140, false, 10-19=0.1
358, 100, 0.7, 0.15, HIGH, 120, 100, 150, false, 10-19=0.1
359, 100, 0.7, 0.15, HIGH, 100, 100, 110, false, 10-19=0.1
360, 100, 0.7, 0.0, HIGH, 120, 100, 150, false, 10-19=0.1
361, 100, 0.7, 0.0, HIGH, 120, 130, 140, false, 10-19=0.1
362, 100, 0.7, 0.0, HIGH, 120, 100, 150, false, 10-19=0.1
363, 100, 0.7, 0.0, HIGH, 100, 100, 110, false, 10-19=0.1
364, 100, 1.3, 0.06, MID, 120, 100, 150, false, 10-19=0.1
365, 100, 1.3, 0.06, MID, 120, 130, 140, false, 10-19=0.1
366, 100, 1.3, 0.06, MID, 120, 100, 150, false, 10-19=0.1
367, 100, 1.3, 0.06, MID, 100, 100, 110, false, 10-19=0.1
368, 100, 1.3, 0.15, MID, 120, 100, 150, false, 10-19=0.1
369, 100, 1.3, 0.15, MID, 120, 130, 140, false, 10-19=0.1
370, 100, 1.3, 0.15, MID, 120, 100, 150, false, 10-19=0.1
371, 100, 1.3, 0.15, MID, 100, 100, 110, false, 10-19=0.1
372, 100, 1.3, 0.06, MID, 120, 100, 150, false, 10-19=0.1
373, 100, 1.3, 0.06, MID, 120, 130, 140, false, 10-19=0.1
374, 100, 1.3, 0.06, MID, 120, 100, 150, false, 10-19=0.1
375, 100, 1.3, 0.06, MID, 100, 100, 110, false, 10-19=0.1
376, 100, 1.0, 0.0, MID, 120, 100, 150, false, 10-19=0.1
377, 100, 1.0, 0.0, MID, 120, 130, 140, false, 10-19=0.1
378, 100, 1.0, 0.0, MID, 120, 100, 150, false, 10-19=0.1
379, 100, 1.0, 0.0, MID, 100, 100, 110, false, 10-19=0.1
380, 100, 0.7, 0.06, MID, 120, 100, 150, false, 10-19=0.1
381, 100, 0.7, 0.06, MID, 120, 130, 140, false, 10-19=0.1

382, 100, 0.7, 0.06, MID, 120, 100, 150, false, 10-19=0.1
383, 100, 0.7, 0.06, MID, 100, 100, 110, false, 10-19=0.1
384, 100, 0.7, 0.15, MID, 120, 100, 150, false, 10-19=0.1
385, 100, 0.7, 0.15, MID, 120, 130, 140, false, 10-19=0.1
386, 100, 0.7, 0.15, MID, 120, 100, 150, false, 10-19=0.1
387, 100, 0.7, 0.15, MID, 100, 100, 110, false, 10-19=0.1
388, 100, 0.7, 0.0, MID, 120, 100, 150, false, 10-19=0.1
389, 100, 0.7, 0.0, MID, 120, 130, 140, false, 10-19=0.1
390, 100, 0.7, 0.0, MID, 120, 100, 150, false, 10-19=0.1
391, 100, 0.7, 0.0, MID, 100, 100, 110, false, 10-19=0.1
392, 100, 1.3, 0.06, LOW, 120, 100, 150, false, 10-19=0.1
393, 100, 1.3, 0.06, LOW, 120, 130, 140, false, 10-19=0.1
394, 100, 1.3, 0.06, LOW, 120, 100, 150, false, 10-19=0.1
395, 100, 1.3, 0.06, LOW, 100, 100, 110, false, 10-19=0.1
396, 100, 1.3, 0.15, LOW, 120, 100, 150, false, 10-19=0.1
397, 100, 1.3, 0.15, LOW, 120, 130, 140, false, 10-19=0.1
398, 100, 1.3, 0.15, LOW, 120, 100, 150, false, 10-19=0.1
399, 100, 1.3, 0.15, LOW, 100, 100, 110, false, 10-19=0.1
400, 100, 1.3, 0.06, LOW, 120, 100, 150, false, 10-19=0.1
401, 100, 1.3, 0.06, LOW, 120, 130, 140, false, 10-19=0.1
402, 100, 1.3, 0.06, LOW, 120, 100, 150, false, 10-19=0.1
403, 100, 1.3, 0.06, LOW, 100, 100, 110, false, 10-19=0.1
404, 100, 1.0, 0.0, LOW, 120, 100, 150, false, 10-19=0.1
405, 100, 1.0, 0.0, LOW, 120, 130, 140, false, 10-19=0.1
406, 100, 1.0, 0.0, LOW, 120, 100, 150, false, 10-19=0.1
407, 100, 1.0, 0.0, LOW, 100, 100, 110, false, 10-19=0.1
408, 100, 0.7, 0.06, LOW, 120, 100, 150, false, 10-19=0.1
409, 100, 0.7, 0.06, LOW, 120, 130, 140, false, 10-19=0.1
410, 100, 0.7, 0.06, LOW, 120, 100, 150, false, 10-19=0.1
411, 100, 0.7, 0.06, LOW, 100, 100, 110, false, 10-19=0.1
412, 100, 0.7, 0.15, LOW, 120, 100, 150, false, 10-19=0.1
413, 100, 0.7, 0.15, LOW, 120, 130, 140, false, 10-19=0.1
414, 100, 0.7, 0.15, LOW, 120, 100, 150, false, 10-19=0.1
415, 100, 0.7, 0.15, LOW, 100, 100, 110, false, 10-19=0.1
416, 100, 0.7, 0.0, LOW, 120, 100, 150, false, 10-19=0.1
417, 100, 0.7, 0.0, LOW, 120, 130, 140, false, 10-19=0.1
418, 100, 0.7, 0.0, LOW, 120, 100, 150, false, 10-19=0.1
419, 100, 0.7, 0.0, LOW, 100, 100, 110, false, 10-19=0.1
420, 100, 1.3, 0.06, HIGH, 120, 100, 150, false, 20-29=0.1
421, 100, 1.3, 0.06, HIGH, 120, 130, 140, false, 20-29=0.1
422, 100, 1.3, 0.06, HIGH, 120, 100, 150, false, 20-29=0.1
423, 100, 1.3, 0.06, HIGH, 100, 100, 110, false, 20-29=0.1
424, 100, 1.3, 0.15, HIGH, 120, 100, 150, false, 20-29=0.1
425, 100, 1.3, 0.15, HIGH, 120, 130, 140, false, 20-29=0.1
426, 100, 1.3, 0.15, HIGH, 120, 100, 150, false, 20-29=0.1
427, 100, 1.3, 0.15, HIGH, 100, 100, 110, false, 20-29=0.1
428, 100, 1.3, 0.06, HIGH, 120, 100, 150, false, 29=1.0
429, 100, 1.3, 0.06, HIGH, 120, 130, 140, false, 29=1.0

430, 100, 1.3, 0.06, HIGH, 120, 100, 150, false, 29=1.0
431, 100, 1.3, 0.06, HIGH, 100, 100, 110, false, 29=1.0
432, 100, 1.0, 0.0, HIGH, 120, 100, 150, false, 20-29=0.1
433, 100, 1.0, 0.0, HIGH, 120, 130, 140, false, 20-29=0.1
434, 100, 1.0, 0.0, HIGH, 120, 100, 150, false, 20-29=0.1
435, 100, 1.0, 0.0, HIGH, 100, 100, 110, false, 20-29=0.1
436, 100, 0.7, 0.06, HIGH, 120, 100, 150, false, 20-29=0.1
437, 100, 0.7, 0.06, HIGH, 120, 130, 140, false, 20-29=0.1
438, 100, 0.7, 0.06, HIGH, 120, 100, 150, false, 20-29=0.1
439, 100, 0.7, 0.06, HIGH, 100, 100, 110, false, 20-29=0.1
440, 100, 0.7, 0.15, HIGH, 120, 100, 150, false, 20-29=0.1
441, 100, 0.7, 0.15, HIGH, 120, 130, 140, false, 20-29=0.1
442, 100, 0.7, 0.15, HIGH, 120, 100, 150, false, 20-29=0.1
443, 100, 0.7, 0.15, HIGH, 100, 100, 110, false, 20-29=0.1
444, 100, 0.7, 0.0, HIGH, 120, 100, 150, false, 20-29=0.1
445, 100, 0.7, 0.0, HIGH, 120, 130, 140, false, 20-29=0.1
446, 100, 0.7, 0.0, HIGH, 120, 100, 150, false, 20-29=0.1
447, 100, 0.7, 0.0, HIGH, 100, 100, 110, false, 20-29=0.1
448, 100, 1.3, 0.06, MID, 120, 100, 150, false, 20-29=0.1
449, 100, 1.3, 0.06, MID, 120, 130, 140, false, 20-29=0.1
450, 100, 1.3, 0.06, MID, 120, 100, 150, false, 20-29=0.1
451, 100, 1.3, 0.06, MID, 100, 100, 110, false, 20-29=0.1
452, 100, 1.3, 0.15, MID, 120, 100, 150, false, 20-29=0.1
453, 100, 1.3, 0.15, MID, 120, 130, 140, false, 20-29=0.1
454, 100, 1.3, 0.15, MID, 120, 100, 150, false, 20-29=0.1
455, 100, 1.3, 0.15, MID, 100, 100, 110, false, 20-29=0.1
456, 100, 1.3, 0.06, MID, 120, 100, 150, false, 20-29=0.1
457, 100, 1.3, 0.06, MID, 120, 130, 140, false, 20-29=0.1
458, 100, 1.3, 0.06, MID, 120, 100, 150, false, 20-29=0.1
459, 100, 1.3, 0.06, MID, 100, 100, 110, false, 20-29=0.1
460, 100, 1.0, 0.0, MID, 120, 100, 150, false, 20-29=0.1
461, 100, 1.0, 0.0, MID, 120, 130, 140, false, 20-29=0.1
462, 100, 1.0, 0.0, MID, 120, 100, 150, false, 20-29=0.1
463, 100, 1.0, 0.0, MID, 100, 100, 110, false, 20-29=0.1
464, 100, 0.7, 0.06, MID, 120, 100, 150, false, 20-29=0.1
465, 100, 0.7, 0.06, MID, 120, 130, 140, false, 20-29=0.1
466, 100, 0.7, 0.06, MID, 120, 100, 150, false, 20-29=0.1
467, 100, 0.7, 0.06, MID, 100, 100, 110, false, 20-29=0.1
468, 100, 0.7, 0.15, MID, 120, 100, 150, false, 20-29=0.1
469, 100, 0.7, 0.15, MID, 120, 130, 140, false, 20-29=0.1
470, 100, 0.7, 0.15, MID, 120, 100, 150, false, 20-29=0.1
471, 100, 0.7, 0.15, MID, 100, 100, 110, false, 20-29=0.1
472, 100, 0.7, 0.0, MID, 120, 100, 150, false, 20-29=0.1
473, 100, 0.7, 0.0, MID, 120, 130, 140, false, 20-29=0.1
474, 100, 0.7, 0.0, MID, 120, 100, 150, false, 20-29=0.1
475, 100, 0.7, 0.0, MID, 100, 100, 110, false, 20-29=0.1
476, 100, 1.3, 0.06, LOW, 120, 100, 150, false, 20-29=0.1
477, 100, 1.3, 0.06, LOW, 120, 130, 140, false, 20-29=0.1

478, 100, 1.3, 0.06, LOW, 120, 100, 150, false, 20-29=0.1
479, 100, 1.3, 0.06, LOW, 100, 100, 110, false, 20-29=0.1
480, 100, 1.3, 0.15, LOW, 120, 100, 150, false, 20-29=0.1
481, 100, 1.3, 0.15, LOW, 120, 130, 140, false, 20-29=0.1
482, 100, 1.3, 0.15, LOW, 120, 100, 150, false, 20-29=0.1
483, 100, 1.3, 0.15, LOW, 100, 100, 110, false, 20-29=0.1
484, 100, 1.3, 0.06, LOW, 120, 100, 150, false, 20-29=0.1
485, 100, 1.3, 0.06, LOW, 120, 130, 140, false, 20-29=0.1
486, 100, 1.3, 0.06, LOW, 120, 100, 150, false, 20-29=0.1
487, 100, 1.3, 0.06, LOW, 100, 100, 110, false, 20-29=0.1
488, 100, 1.0, 0.0, LOW, 120, 100, 150, false, 20-29=0.1
489, 100, 1.0, 0.0, LOW, 120, 130, 140, false, 20-29=0.1
490, 100, 1.0, 0.0, LOW, 120, 100, 150, false, 20-29=0.1
491, 100, 1.0, 0.0, LOW, 100, 100, 110, false, 20-29=0.1
492, 100, 0.7, 0.06, LOW, 120, 100, 150, false, 20-29=0.1
493, 100, 0.7, 0.06, LOW, 120, 130, 140, false, 20-29=0.1
494, 100, 0.7, 0.06, LOW, 120, 100, 150, false, 20-29=0.1
495, 100, 0.7, 0.06, LOW, 100, 100, 110, false, 20-29=0.1
496, 100, 0.7, 0.15, LOW, 120, 100, 150, false, 20-29=0.1
497, 100, 0.7, 0.15, LOW, 120, 130, 140, false, 20-29=0.1
498, 100, 0.7, 0.15, LOW, 120, 100, 150, false, 20-29=0.1
499, 100, 0.7, 0.15, LOW, 100, 100, 110, false, 20-29=0.1
500, 100, 0.7, 0.0, LOW, 120, 100, 150, false, 20-29=0.1
501, 100, 0.7, 0.0, LOW, 120, 130, 140, false, 20-29=0.1
502, 100, 0.7, 0.0, LOW, 120, 100, 150, false, 20-29=0.1
503, 100, 0.7, 0.0, LOW, 100, 100, 110, false, 20-29=0.1

This input data serve to describe the 504 items in such a manner that they are each intended to test the model in the following way:

0: An early item (scheduled with confidence), contractor reasonably confidently expects the final quantity to be 30% more than that captured in the Bills of Quantities, it falls into a workgroup that has a high expected escalation (12%) of which forecast the contractor is relatively confident; there are 100 units in the Bills of Quantities and the contractor estimates their cost to be R110.00 per unit. Prices of less than R100.00 are considered to be at (16%) risk of being unacceptable and, similarly, prices of more than R150.00 are likely unacceptable.

1: Ditto, except that the client is likely to only find prices between R130 and R140 to be acceptable.

2: Same as Item 0 (to test whether the model will price them the same)

3: Ditto, but with a narrow range of acceptable prices (R100 – R110), relatively low compared to the contractor's estimated cost (R110).

4: Same as Items 0 and 2 except that the estimated unit cost is R100 and the contractor is far less confident of the estimate of the quantity variation.

5: Same as Item 1 except that, as with Item 4, the contractor is not confident of the estimate of the quantity variation.

6: Same as Item 4 except that the estimated cost is marginally higher (R105).

7: Ditto, except that the band of acceptable prices is again narrow (R100-R110) which is also worth comparing as being similar to Item 3.

8: Same as Item 0, except that this item is scheduled to be built even earlier – and be built quickly, all in the second month of the project.

9: Similarly, the same as Item 1 except that this item is scheduled to be built even earlier – and be built quickly, all in the second month of the project.

10: Similarly, the same as Item 0 except that this item is scheduled to be built all in the second month of the project.

11: Similarly, the same as Item 7 except that this item is scheduled to be built all in the second month of the project, and the contractor more confident of the expected increase in the quantity that is to be built.

12: The same as Item 0 except the contractor is certain that the quantity built will remain as specified in the Bills of Quantities.

13: Similarly, this item is otherwise the same as Item 1 although the estimated cost is R110 and not R100.

14: This item is the same as Item 12 except that the estimated cost is R120 and not R100.

15: This item is similar to Items 3 and 7 except that the estimated cost is R100 also that the contractor is certain that the quantity built will remain as specified in the Bills of Quantities.

16: This item is similar to Item 12 except that the estimated cost is R120 and also that the contractor thinks it fairly likely that the final quantity will be 30% less than in the Bills of Quantities.

17: This item is similar to Items 1, 5 and 13 except that the estimated cost is R120 and also that the contractor thinks it fairly likely that the final quantity will be 30% less than in the Bills of Quantities.

18: This item is the same as Item 16.

19: This item is similar to Item 15 except that the contractor thinks it fairly likely that the final quantity will be 30% less than in the Bills of Quantities.

20: Similar to Items 16 and 18 except that the contractor is less certain that the final quantity will be 30% less than in the Bills of Quantities.

21: The same as Item 17 except that the contractor is less certain that the final quantity will be 30% less than in the Bills of Quantities.

22: The same as Item 20 except that the estimated unit cost is R110 and not R120.

23: The same as Item 19 except that the contractor is less certain that the final quantity will be 30% less than in the Bills of Quantities.

24: The same as Item 18 except that the contractor is absolutely certain that the final quantity will be 30% less than in the Bills of Quantities.

25: The same as Item 21 except that the contractor is absolutely certain that the final quantity will be 30% less than in the Bills of Quantities.

26: The same as Item 24.

27: The same as Item 23 except that the contractor is absolutely certain that the final quantity will be 30% less than in the Bills of Quantities.

28: The same as Item 0 except that the estimated cost is R120 and not R110 and the item is part of a workgroup that has an average rate of expected escalation and not a high rate.

29: The same as Item 1, ditto.

30: The same as Item 28.

31: The same as Item 3 except that the estimated cost is R100 and not R110 and the item is part of a workgroup that has an average rate of expected escalation and not a high rate.

32: The same as Item 20 except the item is part of a workgroup that has an average rate of expected escalation and not a high rate, and the contractor expects variations will increase the final quantity built and not decrease it.

33: The same as Item 29 except that the contractor is less confident of the estimates of variations.

34: The same as Item 30, ditto.

35: The same as Item 31, ditto.

- 36: The same as Item 28.
- 37: The same as Item 29.
- 38: The same as Item 30.
- 39: The same as Item 31.
- 40: The same as Item 14, except that the item is part of a workgroup that has an average rate of expected escalation and not a high rate.
- 41: The same as Item 13, except that the unit cost is R120 and not R110 and that the item is part of a workgroup that has an average rate of expected escalation and not a high rate.
- 42: The same as Item 40.
- 43: The same as Item 15, except that the item is part of a workgroup that has an average rate of expected escalation and not a high rate.
- 44: The same as Item 18, ditto.
- 45: The same as Item 17, ditto.
- 46: The same as Item 44.
- 47: The same as Item 43, except that the contractor expects variations to the effect that the final quantity will be 30% less than in the Bills of Quantities, of which he is reasonably certain.
- 48: Same as Item 46, except that the contractor is less certain of the likelihood of the variation orders.
- 49: Same as Item 45, ditto.
- 50: Same as Item 48.
- 51: Same as Item 47, except that the contractor is less certain of the likelihood of the variation orders.
- 52: Same as Item 50, except that the contractor is absolutely certain of their expectation of variation orders.

53: Same as Item 49, ditto.

54: Same as Item 52.

55: Same as Item 51, except that the contractor is absolutely certain of their expectation of variation orders.

56-83: Same as Items 28-55 respectively, except that this item falls belong to a workgroup that has a low expected rate of escalation.

84-91: Same as Items 0-7 respectively, except that this item is scheduled to be built over the duration of the middle-third of the overall project.

92-95: Same as Items 8-11 respectively, except that this item is scheduled to be built in the one month halfway through the construction of the overall project.

96-167: Same as Items 12-83 respectively, except that this item is scheduled to be built over the duration of the middle-third of the overall project.

168-175: Same as Items 0-7 and 84-91 respectively, except that this item is scheduled to be built over the duration of the last-third of the overall project.

176-179: Same as Items 8-11 and 92-95 respectively, except that this item is scheduled to be built in the last month of the project.

180-251: Same as Items 12-83 and 96-167 respectively, except that this item is scheduled to be built over the duration of the last-third of the overall project.

252-503: Same as Items 0-251 respectively, except that with all these items the contractor is not as certain of the scheduling as he/she is with the former group of items.

APPENDIX C. TEST RESULTS

This appendix presents a tiny fraction of the amount of data produced by the system in this same manner. If all this data were printed, it would amount to many thousands of pages.

This data contains the following fields of information:

- a reference to the iteration
- the amount of reward generated from this pricing scenario
- the amount of risk, ditto
- the value of utility, ditto

(The first three iterations are special test cases in which it is not appropriate to measure the utility.)

```
0, 5193081, 605469, 0.0
1, 6642313, 648812, 0.0
2, 5913542, 147376, 0.0
3, 5950630, 258974, 4437.0527
4, 6006796, 74761, 15610.713
5, 5924198, 313583, 1295.2406
6, 5961782, 294542, 5134.8276
7, 5982144, 215779, 8413.034
8, 6014881, 306075, 9584.956
9, 6019512, 367678, 8361.63
10, 5998339, 344908, 7371.4644
11, 6012852, 57001, 17474.47
12, 5980918, 254603, 7579.921
13, 5997399, 220435, 9934.121
14, 6012852, 57001, 17474.47
15, 6005916, 153630, 12595.879
16, 5994205, 185348, 10391.61
17, 5998666, 180852, 11007.707
18, 5997591, 191880, 10616.291
19, 5988319, 233836, 8712.873
20, 6004957, 133073, 13110.495
21, 5994030, 226422, 9453.295
```

22, 6013570, 427083, 6267.318
23, 6020525, 380445, 8076.447
24, 6006883, 300456, 9044.109
25, 6006062, 179418, 11883.755
26, 6015091, 237151, 11319.868
27, 6012857, 56991, 17475.85
28, 6012852, 57001, 17474.47
29, 6012274, 370410, 7786.8267
30, 6012601, 445490, 5602.1147
31, 5991493, 473454, 3640.5415
32, 5994060, 339104, 7162.1426
33, 6012857, 56991, 17475.85
34, 6012836, 56995, 17472.404
35, 6012839, 56986, 17473.443
36, 6012840, 56596, 17496.965
37, 6011338, 56549, 17266.598
38, 5944774, 286010, 3576.7942
39, 5981576, 208516, 8489.531
40, 6009386, 412525, 6466.1743
41, 6004004, 168517, 11945.488
42, 6008847, 178976, 12210.357
43, 5979159, 263387, 7255.5557
44, 6015890, 386014, 7615.1724
45, 6003196, 282193, 9131.213
46, 5971288, 232958, 6954.1924
47, 5974654, 240586, 7183.877
48, 6010793, 317174, 8981.024
49, 5984477, 199418, 8989.765
50, 6005547, 185388, 11666.105
51, 6040742, 309851, 11594.073
52, 5925309, 176817, 1947.3417
53, 6003624, 307479, 8610.395
54, 6017242, 297389, 9998.399
55, 5960480, 246681, 5616.3804
56, 5917995, 312371, 602.93506
57, 5988172, 205649, 9269.428
58, 5999218, 181693, 11049.228
59, 6010620, 138886, 13628.749
60, 5991663, 211086, 9532.51
61, 5989104, 260198, 8276.325
62, 5910851, 282578, -1220.7257

APPENDIX D. SYSTEM FEEDBACK

This appendix presents a tiny fraction of the amount of data produced by the system in this same manner. If all this data were printed, it would amount to several hundred pages.

Improvement (due to item pricing priority sequence A) to 4437.052806405739 after 3 efforts, increasing the reward to 5950629 (a 52% improvement on the profit from a balanced bid) whilst reducing the total risk to 258974 (riskReaction = 130796, riskOfBeingWrong = 28179 & riskRejection = 99998)

-1+0-0+0-1+1-0+1-1+1-1+0-0+0...
+f+f+f+f+f+f+f+f+f+f+f+f+f+f+f..

Improvement (due to item pricing priority sequence D) to 15610.713190934639 after 4 efforts, increasing the reward to 6006796 (a 130% improvement on the profit from a balanced bid) whilst reducing the total risk to 74760 (riskReaction = 4704, riskOfBeingWrong = 28410 & riskRejection = 41645)

-0+2-0+0-2+2-0+2-0+1-1+2-0+0-1+1-0+2-0+2-0+2-0+1-0+0-0...
+j+j+j+j+j+j+j+j+f+f+j+j+f+f+j+j+j+j+j+j h h h h h...
Improvement (due to item pricing priority sequence E241) to 17474.470264835185 after 11 efforts, increasing the reward to 6012851 (a 138% improvement on the profit from a balanced bid) whilst reducing the total risk to 57000 (riskReaction = 5601, riskOfBeingWrong = 28438 & riskRejection = 22960)

-8+5-1+1-9+8-2+7-6+5-1+7-0+5-4+4-2+7-3+6-0+3-8+0-5+2-6+7...
+g-g-g-g-g-g...
Improvement (due to item pricing priority sequence C) to 17475.850384954552 after 27 efforts, increasing the reward to 6012857 (a 138% improvement on the profit from a balanced bid) whilst reducing the total risk to 56991 (riskReaction = 5592, riskOfBeingWrong = 28437 & riskRejection = 22960)

-2-9+7-1-3-3-1-2-7+6+0-5+1-3+3-9+1-5-9+4-4+3+2-3...
+g+g+g+g+g+g+g+g+g f-g-g-g-g-g-g-a-g-g-g-g-g-a...

Improvement (due to item pricing priority sequence A) to 17496.965589301093 after 36 efforts, increasing the reward to 6012839 (a 138% improvement on the profit from a balanced bid) whilst reducing the total risk to 56596(riskReaction = 5534, riskOfBeingWrong = 28437 & riskRejection = 22623)

+5-5+8+5-7+9-8+7-5-3+5+6-2+9-8+0+5-3-3+8-3-2-8-8+9+5+0...
 e e+f+f e e e e e-f+f+f+f+f e+f+f e-f-f-f-f-f e e e-f...
 Improvement (due to item pricing priority sequence D) to 17508.607479864142 after 76 efforts, increasing the reward to 6012905 (a 138% improvement on the profit from a balanced bid) whilst reducing the total risk to 56571(riskReaction = 5510, riskOfBeingWrong = 28437 & riskRejection = 22623)

-4-4-5-3-9-7-6-6-2+2-9+9+0+3-2-3-3-5+3-7+4-4+4+2+5+3+4...
 +j+j+j+j+j-j-j+j+j-j h-j-j+j-j+j-j-j-j-j-j h-j-j...
 Improvement (due to item pricing priority sequence E137) to 17512.43667075178 after 86 efforts, increasing the reward to 6012880 (a 138% improvement on the profit from a balanced bid) whilst reducing the total risk to 56444(riskReaction = 5383, riskOfBeingWrong = 28437 & riskRejection = 22623)

-6-5+7+5-6
 +f+f+f+f+f
 Improvement (due to item pricing priority sequence A) to 17514.364388233993 after 87 efforts, increasing the reward to 6012931 (a 138% improvement on the profit from a balanced bid) whilst reducing the total risk to 56544(riskReaction = 5483, riskOfBeingWrong = 28437 & riskRejection = 22623)

+9-6+7-6+6+2-6
 +f-f e-f+f e-f
 Improvement (due to item pricing priority sequence D) to 17516.13128856958 after 88 efforts, increasing the reward to 6012939 (a 138% improvement on the profit from a balanced bid) whilst reducing the total risk to 56534(riskReaction = 5473, riskOfBeingWrong = 28437 & riskRejection = 22623)

+1-2+9+6+3-7+2-3-5+1+3-2-6-6+3-6+5+5-7-9-6-1+4-5-3-3-6-7...
 +h-h g g g g g-h-h g+h+j+j+j-h-h-h-j+h+h g-h-h+j+j-j-h-h...
 Improvement (due to item pricing priority sequence E125) to 17516.286975221807 after 170 efforts, increasing the reward to 6012937 (a 138% improvement on the profit from a

balanced bid) whilst reducing the total risk to
56527(riskReaction = 5466, riskOfBeingWrong = 28437 &
riskRejection = 22623)
-4+9-4+2-2+9-4+9+6-8+0+1-3-3-7-9-6-3+2-7+8-7-3+0-8...
+h+h-h-h-h+h+h-h-h-h g+h+h-h-h+h-h-h-h-h g g+h+h+j...

University of Cape Town

APPENDIX E. ITEM PRICES

This appendix presents the prices and corresponding mark-ups that the system has identified for the hypothetical project as best suited to this contractor (considering their risk profile).

0	R138.92	26.29%
1	R137.50	37.50%
2	R138.44	25.85%
3	R107.72	-2.07%
4	R138.42	38.42%
5	R137.46	37.46%
6	R139.20	32.57%
7	R108.09	2.94%
8	R139.02	26.38%
9	R137.54	37.54%
10	R139.03	26.39%
11	R108.06	2.91%
12	R135.92	23.56%
13	R133.26	21.15%
14	R133.75	11.46%
15	R103.97	3.97%
16	R109.60	-8.67%
17	R132.12	10.10%
18	R109.63	-8.64%
19	R102.48	2.48%
20	R112.33	-6.39%
21	R132.40	10.33%
22	R109.64	-0.33%
23	R102.28	2.28%
24	R112.51	-6.24%
25	R132.49	10.41%
26	R112.39	-6.34%
27	R102.49	2.49%
28	R139.73	16.44%
29	R137.58	14.65%
30	R138.69	15.58%
31	R108.09	8.09%
32	R139.82	16.52%
33	R137.60	14.67%
34	R139.58	16.32%
35	R108.07	8.07%

36	R137.80	14.83%
37	R137.55	14.63%
38	R139.44	16.20%
39	R108.10	8.10%
40	R134.46	12.05%
41	R133.07	10.89%
42	R123.31	2.76%
43	R103.65	3.65%
44	R109.88	-8.43%
45	R132.49	10.41%
46	R110.43	-7.98%
47	R102.41	2.41%
48	R111.96	-6.70%
49	R131.98	9.98%
50	R109.60	-8.67%
51	R101.98	1.98%
52	R109.60	-8.67%
53	R132.49	10.41%
54	R112.40	-6.33%
55	R102.28	2.28%
56	R138.97	15.81%
57	R137.51	14.59%
58	R138.88	15.73%
59	R107.99	7.99%
60	R140.30	16.92%
61	R137.55	14.63%
62	R139.11	15.93%
63	R108.01	8.01%
64	R138.46	15.38%
65	R137.58	14.65%
66	R140.22	16.85%
67	R108.09	8.09%
68	R112.69	-6.09%
69	R132.49	10.41%
70	R112.42	-6.32%
71	R102.58	2.58%
72	R109.60	-8.67%
73	R131.92	9.93%
74	R109.43	-8.81%
75	R102.10	2.10%
76	R109.59	-8.68%
77	R132.48	10.40%
78	R109.59	-8.68%
79	R102.36	2.36%
80	R109.27	-8.94%
81	R132.27	10.23%
82	R112.40	-6.33%
83	R102.34	2.34%

84	R140.53	17.11%
85	R137.51	14.59%
86	R140.13	16.78%
87	R108.09	8.09%
88	R139.98	16.65%
89	R137.52	14.60%
90	R140.42	17.02%
91	R108.00	8.00%
92	R140.34	16.95%
93	R137.52	14.60%
94	R140.23	16.86%
95	R108.12	8.12%
96	R137.38	14.48%
97	R137.37	14.48%
98	R132.38	10.32%
99	R104.49	4.49%
100	R112.38	-6.35%
101	R132.48	10.40%
102	R109.64	-8.63%
103	R102.48	2.48%
104	R109.58	-8.68%
105	R132.48	10.40%
106	R110.50	-7.92%
107	R102.48	2.48%
108	R111.34	-7.22%
109	R132.53	10.44%
110	R109.58	-8.68%
111	R102.49	2.49%
112	R140.11	16.76%
113	R137.55	14.63%
114	R139.42	16.18%
115	R107.97	7.97%
116	R140.41	17.01%
117	R137.60	14.67%
118	R138.70	15.58%
119	R108.09	8.09%
120	R140.40	17.00%
121	R137.53	14.61%
122	R140.42	17.02%
123	R108.07	8.07%
124	R137.46	14.55%
125	R136.52	13.77%
126	R137.33	14.44%
127	R106.49	6.49%
128	R112.44	-6.30%
129	R132.28	10.23%
130	R109.76	-8.53%
131	R102.48	2.48%

132	R112.17	-6.53%
133	R132.48	10.40%
134	R112.36	-6.37%
135	R101.92	1.92%
136	R110.05	-8.29%
137	R132.48	10.40%
138	R112.21	-6.49%
139	R102.49	2.49%
140	R138.47	15.39%
141	R137.51	14.59%
142	R138.48	15.40%
143	R107.82	7.82%
144	R139.96	16.63%
145	R137.54	14.62%
146	R138.62	15.52%
147	R107.51	7.51%
148	R138.14	15.12%
149	R137.51	14.59%
150	R139.70	16.42%
151	R108.03	8.03%
152	R112.82	-5.98%
153	R132.48	10.40%
154	R112.43	-6.31%
155	R102.48	2.48%
156	R109.51	-8.74%
157	R132.48	10.40%
158	R112.26	-6.45%
159	R102.48	2.48%
160	R110.44	-7.97%
161	R132.17	10.14%
162	R109.80	-8.50%
163	R101.94	1.94%
164	R109.60	-8.67%
165	R132.29	10.24%
166	R111.01	-7.49%
167	R102.12	2.12%
168	R140.41	17.01%
169	R137.67	14.73%
170	R140.05	16.71%
171	R108.10	8.10%
172	R140.42	17.02%
173	R137.59	14.66%
174	R140.19	16.83%
175	R108.09	8.09%
176	R139.03	15.86%
177	R137.50	14.58%
178	R140.72	17.27%
179	R107.90	7.90%

180	R137.31	14.43%
181	R137.08	14.23%
182	R123.90	3.25%
183	R102.62	2.62%
184	R112.40	-6.33%
185	R132.48	10.40%
186	R112.38	-6.35%
187	R102.48	2.48%
188	R112.34	-6.38%
189	R132.48	10.40%
190	R112.35	-6.38%
191	R102.48	2.48%
192	R112.39	-6.34%
193	R132.48	10.40%
194	R110.17	-8.19%
195	R102.49	2.49%
196	R140.75	17.29%
197	R137.53	14.61%
198	R140.76	17.30%
199	R108.11	8.11%
200	R140.23	16.86%
201	R137.51	14.59%
202	R140.05	16.71%
203	R108.09	8.09%
204	R140.77	17.31%
205	R137.52	14.60%
206	R140.61	17.18%
207	R108.09	8.09%
208	R135.62	13.02%
209	R135.37	12.81%
210	R125.67	4.73%
211	R103.01	3.01%
212	R112.40	-6.33%
213	R132.48	10.40%
214	R109.97	-8.36%
215	R102.48	2.48%
216	R112.36	-6.37%
217	R132.47	10.39%
218	R112.36	-6.37%
219	R102.48	2.48%
220	R112.39	-6.34%
221	R132.49	10.41%
222	R112.38	-6.35%
223	R102.48	2.48%
224	R137.77	14.81%
225	R136.42	13.68%
226	R137.09	14.24%
227	R107.03	7.03%

228	R138.44	15.37%
229	R137.50	14.58%
230	R138.69	15.58%
231	R107.78	7.78%
232	R137.93	14.94%
233	R137.44	14.53%
234	R138.36	15.30%
235	R107.85	7.85%
236	R112.34	-6.38%
237	R132.48	10.40%
238	R112.38	-6.35%
239	R102.50	2.50%
240	R112.39	-6.34%
241	R132.44	10.37%
242	R109.23	-8.98%
243	R101.85	1.85%
244	R109.24	-8.97%
245	R132.41	10.34%
246	R109.25	-8.96%
247	R102.26	2.26%
248	R109.23	-8.98%
249	R132.41	10.34%
250	R109.20	-9.00%
251	R101.85	1.85%
252	R140.42	17.02%
253	R137.60	14.67%
254	R139.19	15.99%
255	R108.07	8.07%
256	R138.53	15.44%
257	R137.62	14.68%
258	R140.19	16.83%
259	R108.09	8.09%
260	R140.41	17.01%
261	R137.51	14.59%
262	R140.24	16.87%
263	R108.03	8.03%
264	R131.73	9.78%
265	R137.36	14.47%
266	R122.44	2.03%
267	R105.38	5.38%
268	R112.40	-6.33%
269	R132.48	10.40%
270	R110.06	-8.28%
271	R102.43	2.43%
272	R109.92	-8.40%
273	R132.48	10.40%
274	R109.60	-8.67%
275	R102.48	2.48%

276	R109.60	-8.67%
277	R132.39	10.33%
278	R109.60	-8.67%
279	R102.49	2.49%
280	R139.88	16.57%
281	R137.64	14.70%
282	R138.30	15.25%
283	R108.04	8.04%
284	R140.41	17.01%
285	R137.55	14.63%
286	R138.91	15.76%
287	R108.15	8.15%
288	R140.34	16.95%
289	R137.61	14.68%
290	R138.46	15.38%
291	R108.09	8.09%
292	R135.89	13.24%
293	R137.43	14.53%
294	R116.90	-2.58%
295	R102.49	2.49%
296	R112.36	-6.37%
297	R132.47	10.39%
298	R109.38	-8.85%
299	R102.40	2.40%
300	R109.62	-8.65%
301	R132.50	10.42%
302	R112.39	-6.34%
303	R102.48	2.48%
304	R109.63	-8.64%
305	R132.55	10.46%
306	R109.65	-8.62%
307	R102.49	2.49%
308	R139.80	16.50%
309	R137.51	14.59%
310	R138.61	15.51%
311	R108.05	8.05%
312	R139.59	16.33%
313	R137.51	14.59%
314	R138.62	15.52%
315	R108.08	8.08%
316	R138.45	15.38%
317	R137.50	14.58%
318	R140.26	16.88%
319	R107.97	7.97%
320	R112.40	-6.33%
321	R132.49	10.41%
322	R112.42	-6.32%
323	R102.82	2.82%

324	R109.59	-8.68%
325	R132.48	10.40%
326	R112.40	-6.33%
327	R102.23	2.23%
328	R110.51	-7.91%
329	R132.48	10.40%
330	R109.52	-8.73%
331	R101.93	1.93%
332	R109.60	-8.67%
333	R132.45	10.38%
334	R109.57	-8.69%
335	R102.49	2.49%
336	R139.66	16.38%
337	R137.68	14.73%
338	R139.45	16.21%
339	R108.08	8.08%
340	R140.40	17.00%
341	R137.59	14.66%
342	R140.01	16.68%
343	R108.09	8.09%
344	R138.68	15.57%
345	R137.54	14.62%
346	R140.43	17.03%
347	R108.09	8.09%
348	R136.26	13.55%
349	R134.94	12.45%
350	R112.93	-5.89%
351	R106.99	6.99%
352	R112.37	-6.36%
353	R132.48	10.40%
354	R112.38	-6.35%
355	R102.48	2.48%
356	R112.36	-6.37%
357	R132.48	10.40%
358	R112.34	-6.38%
359	R102.50	2.50%
360	R110.26	-8.12%
361	R132.48	10.40%
362	R111.93	-6.72%
363	R102.25	2.25%
364	R140.41	17.01%
365	R137.62	14.68%
366	R139.48	16.23%
367	R108.07	8.07%
368	R139.79	16.49%
369	R137.54	14.62%
370	R140.41	17.01%
371	R108.05	8.05%

372	R139.44	16.20%
373	R137.61	14.68%
374	R140.39	16.99%
375	R108.08	8.08%
376	R128.06	6.72%
377	R134.57	12.14%
378	R122.50	2.08%
379	R104.80	4.80%
380	R112.38	-6.35%
381	R132.48	10.40%
382	R111.73	-6.89%
383	R102.45	2.45%
384	R112.36	-6.37%
385	R132.48	10.40%
386	R112.02	-6.65%
387	R102.18	2.18%
388	R109.59	-8.68%
389	R132.14	10.12%
390	R111.22	-7.32%
391	R102.48	2.48%
392	R138.21	15.18%
393	R137.51	14.59%
394	R138.55	15.46%
395	R107.82	7.82%
396	R138.59	15.49%
397	R137.52	14.60%
398	R139.82	16.52%
399	R107.84	7.84%
400	R138.55	15.46%
401	R137.53	14.61%
402	R138.47	15.39%
403	R107.79	7.79%
404	R111.76	-6.87%
405	R132.48	10.40%
406	R112.40	-6.33%
407	R104.77	4.77%
408	R110.72	-7.73%
409	R132.50	10.42%
410	R109.60	-8.67%
411	R101.85	1.85%
412	R111.59	-7.01%
413	R132.48	10.40%
414	R109.60	-8.67%
415	R101.85	1.85%
416	R109.47	-8.78%
417	R132.46	10.38%
418	R109.61	-8.66%
419	R101.86	1.86%

420	R140.64	17.20%
421	R137.62	14.68%
422	R140.77	17.31%
423	R108.10	8.10%
424	R139.87	16.56%
425	R137.62	14.68%
426	R140.77	17.31%
427	R108.09	8.09%
428	R140.54	17.12%
429	R137.57	14.64%
430	R140.71	17.26%
431	R108.16	8.16%
432	R121.01	0.84%
433	R136.37	13.64%
434	R138.29	15.24%
435	R107.67	7.67%
436	R112.37	-6.36%
437	R132.48	10.40%
438	R112.38	-6.35%
439	R102.48	2.48%
440	R111.10	-7.42%
441	R132.48	10.40%
442	R109.96	-8.37%
443	R102.48	2.48%
444	R112.38	-6.35%
445	R132.48	10.40%
446	R110.90	-7.58%
447	R102.48	2.48%
448	R140.26	16.88%
449	R137.59	14.66%
450	R140.73	17.28%
451	R108.07	8.07%
452	R140.74	17.28%
453	R137.52	14.60%
454	R140.77	17.31%
455	R108.09	8.09%
456	R140.77	17.31%
457	R137.56	14.63%
458	R140.67	17.23%
459	R108.07	8.07%
460	R124.65	3.88%
461	R132.69	10.58%
462	R138.24	15.20%
463	R104.16	4.16%
464	R109.92	-8.40%
465	R132.48	10.40%
466	R112.50	-6.25%
467	R102.48	2.48%

468	R110.88	-7.60%
469	R132.63	10.53%
470	R112.34	-6.38%
471	R102.48	2.48%
472	R112.38	-6.35%
473	R132.48	10.40%
474	R112.43	-6.31%
475	R102.61	2.61%
476	R138.22	15.18%
477	R137.42	14.52%
478	R138.31	15.26%
479	R107.50	7.50%
480	R137.53	14.61%
481	R137.52	14.60%
482	R138.53	15.44%
483	R106.05	6.05%
484	R138.68	15.57%
485	R137.38	14.48%
486	R138.51	15.43%
487	R107.44	7.44%
488	R112.36	-6.37%
489	R132.48	10.40%
490	R112.70	-6.08%
491	R102.48	2.48%
492	R112.41	-6.33%
493	R132.36	10.30%
494	R109.24	-8.97%
495	R101.85	1.85%
496	R109.88	-8.43%
497	R132.36	10.30%
498	R109.33	-8.89%
499	R101.86	1.86%
500	R109.31	-8.91%
501	R132.48	10.40%
502	R109.28	-8.93%
503	R102.48	2.48%