A novel algorithm for analysing gravitational microlensing events

Vinesh Rajpaul

A dissertation presented for the degree of Master of Science
in the Department of Astronomy at the University of Cape Town

Supervisors: Dr J. W. Menzies (SAAO), Dr I. M. Stewart (Bonn) and
A/Prof. P. A. Woudt (UCT)

August 2012

Abstract

Gravitational microlensing is an established technique for detecting extrasolar planets. Very comprehensive models exist for describing microlensing events and their corresponding lightcurves, though unfortunately it is a notoriously difficult and time-consuming task to use these models to interpret microlensing events. Amongst other complicating factors, the models tend to be highly nonlinear, and have enormous parameter spaces that are often fraught with ambiguities and degeneracies. Suffice it to say that modelling ongoing events, in order to forecast possible upcoming features in lightcurves (and thus to inform future observations), is an even more difficult task.

This dissertation presents a new algorithm that was developed to perform autonomous fitting of gravitational microlensing lightcurves. The algorithm combines features of extant evolutionary algorithms with some novel ones, and fares well on the problem of fitting binary lens microlensing lightcurves, as well as on a number of other difficult optimisation problems. Furthermore, the new algorithm is conceptually simple, versatile and robust, and parallelises trivially.

A success rate of about 95% is achieved when using the algorithm to fit synthetic but noisy binary-lens lightcurves, allowing no more than 20 minutes per fit on a desktop computer; this success rate is shown to compare favourably with that of both a conventional (iterated simplex) algorithm and a more state-of-the-art artificial neural network-based approach. Moreover, it is shown that the algorithm is capable – at least, to the extent permitted by the inherent degeneracies and nonlinearities associated with microlensing models – of using incomplete microlensing lightcurves (both real and synthetic) to forecast possible upcoming features.

Further work is required to investigate how the algorithm will fare when faced with more complex and realistic microlensing modelling problems; it is, however, argued here that the use of parallel computing platforms, such as inexpensive graphics processing units, should allow fitting times to be constrained to under an hour, even when dealing with complicated microlensing models. This dissertation therefore provides proof of concept for the use of an evolutionary algorithm as the basis for real-time, autonomous modelling of microlensing events. It is also hoped that this work might stimulate some interest in evolutionary algorithms, and that algorithm presented here might prove useful for solving more general model-fitting and optimisation problems.

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviations

ANN	artificial neural network
CPU	central processing unit
EA	evolutionary algorithm
EMMA	Evolutionary Microlensing Modelling Algorithm
ES	evolution strategy
GA	genetic algorithm
(GP)GPU	(general purpose) graphics processing unit
MAP	maximum *a posteriori*
ML	maximum likelihood
MOA	Microlensing Observations in Astrophysics
NFL	no free lunch
OGLE	Optical Gravitational Lensing Experiment
pGcs	points per GHz per core per second
SBLM	simple binary lens model

Assumed set notation

$\mathbb{C}$	the set of all complex numbers
$\mathbb{N}$	the set of all natural numbers: $\{0, 1, 2, \ldots\}$
$\mathbb{N}^*$	the set of all positive integers: $\{1, 2, 3, \ldots\}$, or $\mathbb{N}\backslash\{0\}$
$\mathbb{R}$	the set of all real numbers
$\mathbb{Z}$	the set of all integers

Acknowledgments

I wish to thank my supervisors – Dr J. W. Menzies, Dr I. M. Stewart, and Prof. P. A. Woudt – for their guidance and support. In particular, I'd like to thank my primary supervisor, Dr Menzies, for agreeing to supervise this project, and for granting me considerable autonomy with respect to the direction taken by my research; for providing funding to support research-related travel and equipment; and of course, for sharing with me his extensive microlensing expertise, and for providing me with a great deal of valuable advice. I wish to thank my co-supervisor, Dr Stewart, for sharing with me his significant experience on all things computational, and for always being willing to provide patient, cheerful counsel, even after moving to distant shores. Finally, I'd like to thank my official supervisor, Prof. Woudt, for helping me out with general dissertation-related matters – both academic and administrative – and for always being on hand to provide friendly encouragement and guidance.

I am grateful to the University of Cape Town (UCT) and the National Research Foundation (NRF) for providing partial financial support for this work. I am also grateful to the members of staff in the Astronomy Department at UCT, as well as at the South African Astronomical Observatory (SAAO), for their support, and for always making me feel at home: it was a real pleasure to work at these two institutions over the course of the past year and a half. I also wish to thank my office-mates and 'colleagues', at both UCT and at SAAO, for providing me with enjoyable and stimulating company.

On a more practical note, I prepared this dissertation using LaTeX, and I must acknowledge Jamie Stevens, whose beautifully-typeset Ph.D. thesis (Stevens, 2005) served as a template for the layout of much of this work. I am also grateful to the developers of the excellent TeamViewer software, which enabled me to configure, run, and monitor – with ease, and from anywhere in the world – all of my computational simulations.

Finally, and on a more personal note, I am indebted to my family and friends for their constant and unconditional love and support. Without them, this work would not have been possible.

Plagiarism declaration

I, Vinesh Rajpaul, know the meaning of plagiarism, and declare that all of the work in the dissertation, save for that which is properly acknowledged, is my own.

Signed:

24 August 2012

1 Introduction

Gravitational microlensing is a well-established technique for detecting extrasolar planets. When a massive foreground object (the lens, e.g. a planet and its host star) passes in front of a distant, background star (the source), the latter is magnified and displays a characteristic microlensing lightcurve. In this way, microlensing allows the lens to be detected on account of its mass, rather than its luminosity, and therefore very faint (even dark), planetary-mass objects can be detected via microlensing. Indeed, microlensing has the potential to yield the most representative statistical sample of Milky Way planets: unlike many complementary techniques used to detect extrasolar planets (exoplanets), it is in principle sensitive enough to detect even very distant, Earth-mass planetary objects at wide separations from their host stars (Bennett & Rhie, 1996; Wambsganss, 2011).

Unfortunately, microlensing events are extremely rare, requiring a very precise alignment between observer, lens and source. As of August 2012, of the several hundred known exoplanets, only around twenty have been discovered by microlensing (see Shvartzvald and Maoz, 2012, and references contained therein). Still, many of these detections constituted have unusually important discoveries in the broader context of exoplanetary science (e.g. Beaulieu et al., 2006; Gaudi et al., 2008; Cassan et al., 2012).

Although very comprehensive models exist for describing microlensing events and their corresponding lightcurves, it is notoriously difficult to use these models to interpret microlensing events, and in spite of significant advances – on computational, mathematical, and algorithmic fronts – made in recent years, modelling microlensing events remains a challenging task. Amongst other complicating factors, microlensing models tend to be highly nonlinear, and have enormous parameter spaces which are often fraught with ambiguities and degeneracies (Dominik, 1999a; Vermaak, 2007). Even the simplest possible microlensing model, viz. that of a point-like source star lensed by a point-mass, poses many nontrivial challenges to microlensing modellers (Dominik, 2008, 2009). Suffice it to say that modelling ongoing events, in order to forecast possible upcoming features in lightcurves (to guide future observations, for example), is an even more challenging task.

This dissertation is intended to provide proof of concept for the use of an evolutionary algorithm as the basis for real-time, autonomous modelling of *ongoing* microlensing events. Accordingly, the dissertation introduces a new metaheuristic algorithm that combines features of extant evolutionary algorithms (including genetic algorithms and evolution strategies) with some novel ones, developed with a view to performing efficient and autonomous fitting of (especially binary-lens) microlensing lightcurves. The new

algorithm is conceptually simple, versatile and robust, and parallelises trivially.

Given the task of fitting complete binary-lens lightcurves, the new algorithm is shown to outperform both a conventional (iterated simplex) algorithm and a more state-of-the-art, artificial neural network-based approach. It is also demonstrated that the new algorithm can perform fits rapidly whilst maintaining a fairly modest computational footprint. Finally, it is demonstrated that the algorithm is capable – at least, to the extent permitted by the inherent degeneracies and nonlinearities associated with microlensing models – of using incomplete microlensing lightcurves to forecast possible upcoming features.

The balance of this dissertation is structured as follows. Chapter 2 gives a brief overview of gravitational microlensing and, in particular, of the binary-lens model used throughout this work. Chapter 3 describes evolutionary algorithms in general, while Chapter 4 describes the nuts and bolts of the newly-developed evolutionary algorithm. Chapter 5 describes the fitting experiments used to assess the algorithm's performance when fitting complete microlensing lightcurves, and chapter 6 describes the experiments used to study the algorithm's success at forecasting features in the lightcurves of *ongoing* microlensing events. Finally, Chapter 7 summarises and concludes.

Chapters 4–6 of this dissertation present original research and results, whilst Chapters 2–3 constitute, for the most part, a review of literature relevant to the results presented in the later chapters. In particular, it is worth noting here that Chapter 3 and Chapters 4–5 of this dissertation are adapted from two separate, peer-reviewed publications, viz. Rajpaul (2011) and Rajpaul (2012), respectively. The results presented in Chapter 6 of this dissertation are also currently (i.e. at the time of submission of this dissertation) being considered by the author for submission for peer-review and possible publication, possibly following some further simulations and analysis. Finally, the author of this dissertation recently co-authored an article (Veller & Rajpaul, 2012), published in a prominent physics journal, which studies evolutionary dynamics and some of its applications in the context of game theory. Because of its mathematical (rather than astrophysical) nature, and to obviate a lengthy, tangential discussion of nonlinear dynamics and game theory, no content from the latter article has been included in this dissertation. Nevertheless, it is interesting to note that some of the research on which this dissertation was founded has already led to two very disparate applications, one astrophysical, and one game theoretical.

2
Gravitational microlensing

In contradistinction to most techniques used to discover extrasolar planets, gravitational microlensing does not rely on the detection of electromagnetic radiation from a planet or its host star; rather, the presence of a planet is inferred solely by its gravitational deflection of light from a more distant source, typically a star. Unfortunately, microlensing events are extremely rare, they cannot be predicted in advance, and planetary microlensing signals – when they do occur – typically last less than a day. To make matters worse, the modelling and interpretation of microlensing lightcurves is often an extremely difficult task!

Nevertheless, microlensing provides a powerful probe of the Galactic exoplanet population, and one which is strongly complementary to other techniques for planet detection; it is particularly sensitive to low-mass planets on wide orbits, and can be used to search for planets orbiting host stars with a broad range of masses and distances from the centre of the Galaxy. Therefore, in spite of its drawbacks, microlensing has very rapidly become one of the most active and exciting topics of research in exoplanetary science.

This chapter focuses solely on gravitational microlensing, with particular emphasis on topics that will prove relevant later in this dissertation. Sec. 2.1 of the chapter provides a general though cursory introduction to gravitational (micro)lensing, and includes some historical notes, a discussion of microlensing's role in detecting exoplanets, and a quick look at contemporary microlensing observational experiments and modelling. Sec. 2.2 introduces a simple microlensing model that describes a point-like source lensed by a single, massive body. Sec. 2.3 develops and discusses, in some detail, a model for the lensing of a point-like source by a *binary* lens – a model that will play a central role throughout most of this dissertation. Sec. 2.4 discusses possible extensions to the binary-lens model considered in the preceding section, and Sec. 2.5 concludes.

2.1 A (very brief) introduction to microlensing

2.1.1 General relativity, and observing the bending of light

The theory of general relativity predicts that any massive body should warp spacetime, and in so doing, distort the geodesic propagation paths taken by photons of any frequency. In fact, the more massive the body, the stronger the resulting distortion

(Einstein, 1916). Because this general relativistic effect amounts to the bending of the paths taken by light, it is referred to as 'gravitational lensing'.

The predicted phenomenon of gravitational lensing was first confirmed during the 1919 solar eclipse, when Sir Arthur Eddington observed that stars close to the solar limb appeared slightly out of position, on account of the paths of their light being distorted, *en route* to the Earth, by the gravitational field of the sun (Dyson et al., 1920).

Einstein realised that it should be possible for astronomical objects other than the sun also to bend light, and that under the correct conditions, one should be able to observe multiple distorted images of a single source. In 1936, he published a paper in which he derived the equations governing the lensing of a background star by a closely-aligned foreground star; he described the appearance of rings and double images, and he computed the amount by which the background star should be magnified during the lensing (Einstein, 1936). However, because of the minuscule angular scales involved, and because of the exceptionally low probability of such an event actually occurring, he concluded that there would be "no great chance" of the effect ever being observed.

In 1937, Fritz Zwicky turned his attention to gravitational lensing on mass and distance scales far grander than those contemplated by Einstein. He concluded that observing the pronounced lensing effects of a galaxy – lensing a background galaxy, say – should have been well within the scope of contemporary observations (Zwicky, 1937).

In fact, it was not until 1979 that Zwicky's predictions were confirmed, with the discovery of the so-called 'Twin QSO', or SBS 0957+561. What appeared initially to be two inexplicably similar quasars was actually a single quasar that had been lensed into two images by a foreground object, viz. a giant elliptical galaxy (Walsh, Carswell & Weymann, 1979). With the discovery of the Twin QSO was born a new field of astronomy, and one that remains, three decades down the line, a very active and important field of research. Over the years, observations of gravitational lensing on galactic scales (usually divided into the regimes of 'strong' and 'weak', depending on whether the formation of multiple images is observed) have facilitated discoveries of the most distant known galaxies, the mapping the dark matter distribution in the universe, and even studies of the expansion of the universe (Schneider et al., 2006).

Gravitational *microlensing* is a special case of gravitational lensing where, because the lens masses are so low, the source images that are formed are separated by the order of a milliarcsecond or less – thus, given the capabilities of modern optical telescopes, the images are unresolvable. Nevertheless, the amplification or brightening of a source during microlensing (due to the presence of the multiple unresolved images) should, in principle, be detectable with high-quality photometry. In fact, the source amplification or magnification should vary as a function of time, on account of relative motion of the lens with respect to the source, which would give rise to a so-called 'lightcurve' for the event.

Following the publication of Einstein's 1936 paper (which involved stellar lenses lensing stellar sources – thus, well within the microlensing regime), the idea of gravitational microlensing lay largely dormant for nearly half a century (see, however Liebes, 1964; Refsdal, 1964, 1966). It was not until the mid-1980s, following a seminal paper by Paczyński (1986), that interest in microlensing was rejuvenated. Paczyński proposed

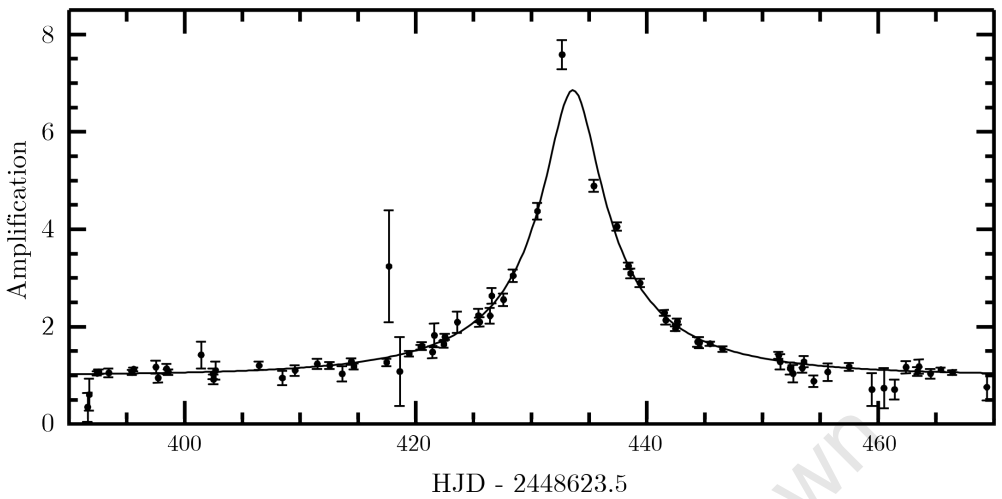

Figure 2.1 – Lightcurve of the first confirmed microlensing event, discovered in 1993. The lensed source was a star in the Large Magellanic Cloud, while the lens in question was an object with an estimated mass of $\sim 0.12\ M_{\odot}$. Source flux from a red passband (630–760 nm), along with $\pm 1\sigma$ errors, is represented by black points, while the smooth solid curve shows the best-fitting microlensing model. Plot produced using data from Alcock et al. (1993).

using microlensing to look for dark matter in the form of massive compact halo objects (MACHOs) in the Galactic halo, by observing background stars in a nearby galaxy.

By the early 1990s, several observational searches were underway, hunting for microlensing events towards the Galactic Bulge and the Large and Small Magellanic Clouds. The first microlensing by an isolated object was observed in 1993 (Alcock et al., 1993; the lightcurve in Fig. 2.1 refers), and many more microlensing events were to be observed in the years that followed. To date, many *thousands* of events have been observed by a number of different collaborations, with the vast majority of observed events involving the lensing action of a single star on a source star observed towards the Galactic Bulge.

Still, Einstein was not too far off the mark when he speculated that microlensing would have "no great chance" of being observed. Indeed, the chance of observing any individual microlensing event is extremely small as it depends on observing a very precise alignment between a foreground and background star – a rare and stochastic event that cannot be predicted in advance. In fact, the alignment required for a microlensing event to be observable is so precise that, even towards the crowded field of the Galactic Bulge (where the high stellar density maximises microlensing event rates), a randomly-chosen source star will be lensed by a foreground star only once every several hundred thousand years (Gaudi, 2012)! Thus, very large numbers of stars must be monitored continuously if one is to have any hope of detecting a microlensing event. To make matters worse, the crowded nature of observed fields increases photometric noise, and necessitates very high angular resolution imaging for unambiguous detection of light from a lens. It is not surprising, then, that it took enormous advances in telescopic resolving capabilities,

imaging technology, and computing power before it started to seem feasible ever to detect a microlensing event.

Since first being observed in 1993, microlensing has found a number of important applications in astronomy and astrophysics, including many applications distinct from those in the strong- and weak-lensing regimes. For example, microlensing has been used to constrain the nature of dark matter, to study limb darkening in distant stars, to constrain the binary stellar population, and to constrain the structure of the Milky Way's disk (Mao, 2012). It has also been proposed as a means to measure stellar rotation, probe the accretion disks of quasars, find dark objects like brown dwarfs and black holes, and more.

Almost without doubt, however, microlensing's 'crowning glory' over the past decade or so has been the discovery of a number of extrasolar planets. This is discussed in more detail below.

2.1.2 Microlensing and exoplanets

Even before the first microlensing events were discovered, Mao & Paczyński (1991) suggested that gravitational lensing by binary lenses could give rise to lightcurves that differ significantly from the simple, symmetric lightcurves produced by single lenses (see Sec. 2.2). In particular, they noted the appearance of extra pairs of images, and the possibility of unusually high source magnifications. They also noted that the probability of observing these distinctive binary signatures would remain nontrivial even if one of the lenses were to be significantly lighter than the other (for example, if the binary were to comprise a solar-mass star and a Jovian-mass companion), provided source magnification could be monitored very closely over time. They thus argued that microlensing could be a fruitful way of searching for exoplanetary systems, and they estimated that a typical Jovian-mass planet would have a ~ 5 to 10% probability of being detected if its parent star were to act as a lens, with a normal star in the Galaxy as a source. They observed that the probability of detecting a low-mass secondary lens would be dependent on the ratio of the secondary's mass to that of the primary lens (the lower the mass ratio, the smaller the perturbations to the standard single-lens lightcurve), and also on various geometrical factors, including the projected orbital radius of the secondary lens, the projected angle at which the source passes the binary system, the alignment of the source, lens and observer, etc.

Microlensing searches for exoplanets began in earnest in 1995, with the first uncontroversial detection of a planet being made in 2003 (Bond et al., 2004). The planet in question had an estimated mass of 2.6 $M_{\rm J}$, and was separated by 4.3 AU from its small host star that lies some 5.8 kpc from Earth. The lightcurve in Fig. 2.2 refers.

Since then, and as of mid-August 2012, microlensing has led to about 20 probable exoplanet detections (16 of which have been published[1]), along with a similar number of promising candidates that await detailed analysis. While the total number of microlensing-based discoveries pales in comparison to the number of exoplanets discov-

[1]For a review and commentary on most of these detections, refer to Bond (2012).

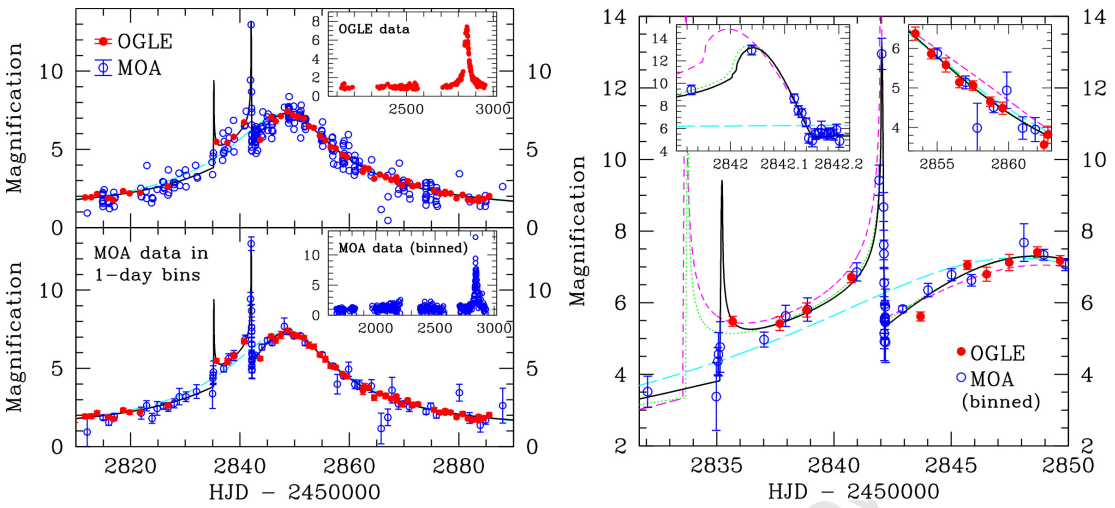

Figure 2.2 – Lightcurve of the event OGLE-2003-BLG-235/MOA-2003-BLG-53 (reproduced from Bond et al., 2004); the solid black curve is the best-fitting binary model for the event. Analysis of this lightcurve led to the first definitive, microlensing-based discovery of an exoplanet.

ered via the radial-velocity/Doppler method and the transit method (see Fig. 2.1 and Table 2.3 – in fact, as of August 2012, the Kepler spacecraft alone has led to the discovery of more than 70 confirmed exoplanets, and more than 2300 exoplanet candidates, via the transit method!), microlensing is nevertheless strongly complementary to the other planet-detection methods.

For example, as suggested in Fig. 2.4, microlensing is the only technique sensitive to very low-mass planets in wide orbits, including orbits at or beyond the ice line (the location in protoplanetary disks where the temperature is cool enough for compounds such as water, ammonia and methane to condense into solid ice grains), a region of considerable importance in some planet-formation theories. Microlensing is also sensitive to a very broad range of host stellar types (and masses), including main-sequence stars, brown dwarfs, and stellar remnants; these host stars may also span a broad range of distances, and will typically be located anywhere from a few hundred parsecs from the sun, to as far out as the Galactic centre, at a distance of some 8.3 kpc. In contrast, most other planet-detection techniques tend to be strongly biased towards more luminous or nearby host stars. Microlensing is also less strongly biased towards finding more massive exoplanets than many other methods, including the Doppler method and the transit method.

Thus, microlensing is uniquely placed to characterise the Galactic planetary population. Many of the microlensing discoveries to date have constituted important discoveries in the broader context of exoplanetary science (e.g. Beaulieu et al., 2006; Gaudi et al., 2008; Cassan et al., 2012), and microlensing surveys have already provided important constraints on the frequency of cold terrestrial planets, giant planets, and Solar System homologues. Recently, it has even been suggested that microlensing has facilitated

Method	Exoplanets	
	NASA[2]	Exoplanet.eu[3]
Transit timing variation	0	?
Astrometry	1	1
Pulsar timing variation	5	15
Eclipse timing variation	6	?
Microlensing	14	16
Imaging	23	31
Transit	220	239
Radial velocity	495	475
Total	764	≥ 777

Table 2.1 – Approximate total number of exoplanet detections, per detection method, according to two different exoplanet catalogues (numbers updated mid-August 2012). The slightly different numbers across the two columns are due to the different inclusion criteria used by the two catalogues (this reflects the fact that there is still no universally-accepted definition of an exoplanet). For example, the NASA catalogue includes only published detections of bodies with masses minimum masses $\leq 30\ M_{\rm J}$, whereas the Exoplanet.eu catalogue uses a stricter mass cut-off of 20 $M_{\rm J}$ (up to 2σ), but does include detections that have been formally announced though not yet documented in peer-reviewed journals. The counting is further complicated by some detections having more corroborating evidence than others.

the detection of a number of free-floating, planetary mass objects in the Galaxy (Sumi et al., 2011; cf., however, Quanz et al., 2012). Microlensing is also the only technique with the potential to detect planets outside the Galaxy: Ingrosso et al. (2009) have already claimed the possible existence of one such planet in M31.

Of course, microlensing is not without its fair share of drawbacks (over and above the already-mentioned problem of microlensing events being rare, unpredictable and transient).

For one, microlensing provides only a snapshot of planetary systems, and unlike many complementary techniques, it does not provide dynamical information such as the orbital period of planets around their host stars. Repeat observations are also usually not possible: a lensing system will typically need to travel several arcseconds or more across the sky before it aligns with a second source star, and since lens motions are characterised by speeds on the order of milliarcseconds per year, it will take hundreds or thousands of years before alignment with another source star is achieved.

Another complicating factor is that with microlensing, physically important quantities can generally only be measured in relative units (see Secs. 2.2 and 2.3): for example, while the *ratio* of masses in a binary system can be measured, absolute masses cannot be determined directly from a microlensing lightcurve. Thus, even after a full microlensing

[2]Data retrieved on 15 August 2012 from the NASA Exoplanet Archive, available online at `http://exoplanetarchive.ipac.caltech.edu`.

[3]Data retrieved on 15 August 2012 from the Extrasolar Planets Encyclopaedia, available online at `http://exoplanet.eu`.

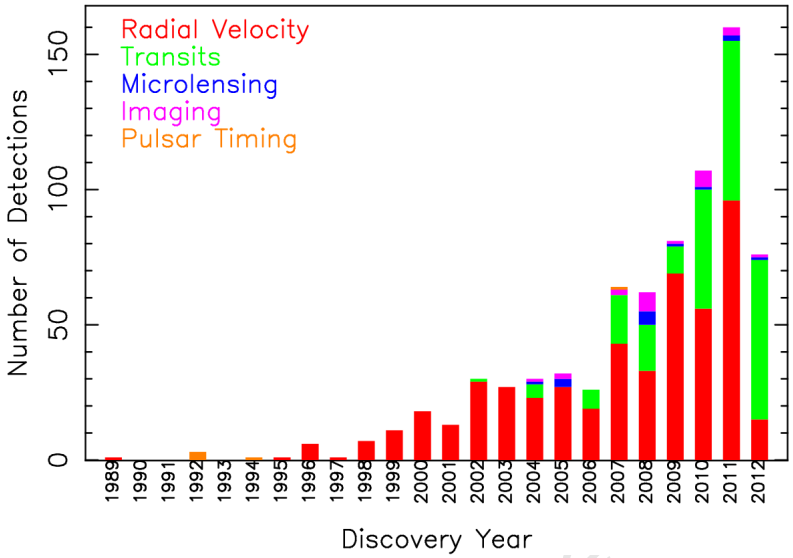

Figure 2.3 – Exoplanet detections by year, and by detection technique; some 764 detections are represented. Plot produced using 15 August 2012 data from the NASA Exoplanet Archive, available online at `http://exoplanetarchive.ipac.caltech.edu`.

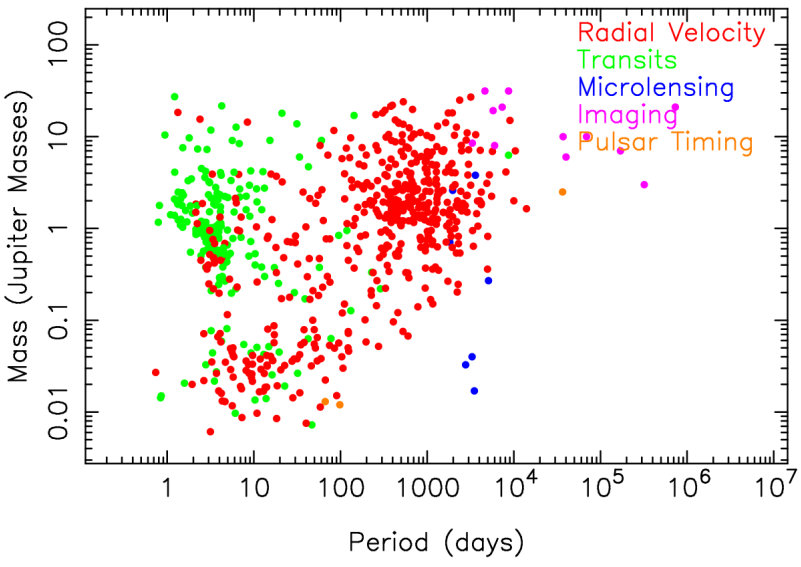

Figure 2.4 – Scatter-plot showing masses and orbital periods of the $\sim$ 764 'known' exoplanets, with colours indicating method of detection. Plot produced using 15 August 2012 data from the NASA Exoplanet Archive, available online at `http://exoplanetarchive.ipac.caltech.edu`.

lightcurve has been obtained, substantial further effort is required to try to characterise a putative planet's host star, which will often be distant (> 1 kpc) and faint ($V \gtrsim 18$).[4] This need has motivated the use of higher-order effects in microlensing events to characterise the properties of planetary hosts (see Sec. 2.4), as well as space-based imaging that can disentangle host star flux from the flux of the dense background fields (Gaudi, 2012). In the absence of a well-characterised host star, one needs to make various assumptions – usually informed by Galactic stellar population statistics – about the nature of the lens system before one can constrain properties such as physical lens masses, projected separations, etc.

Finally, there is a bias towards finding planets separated from their stars by a certain radius known as the 'Einstein radius' (see Sec. 2.2), since planets at or near this separation from their hosts will tend to produce the most pronounced deviations in lightcurves.

For a detailed discussion of the relative merits of the different techniques available for detecting exoplanets, refer e.g. to Udry & Santos (2007) or Mason (2008).

2.1.3 In search of exoplanets: modern microlensing experiments

The transient, unpredictable nature of microlensing events necessitates continuous observation with cadences that are significantly shorter than the timescales of interest. Microlensing events involving stellar-mass lenses have a median timescale of ~ 20 days, but planetary perturbations to these events are characterised by timescales of only a few days or even hours (Griest & Safizadeh, 1998). Thus, while relatively low cadences (on the order of a few days) are sufficient to detect 'primary' events, detecting and accurately characterising planetary perturbations to these events requires cadences of well under an hour. One practical consequence is that, since observational cadences must be much less than one day, longitudinally-distributed observatories are required to ensure 'round-the-clock' coverage of active events.

The rarity of microlensing events, in turn, means that enormous numbers of stars ($\gg 10^6$) must be monitored in order to obtain a realistic yield of primary events. This necessitates the observation of dense stellar fields, which, in turn, means accurate crowded field photometry is essential (Woźniak, 2000). Fortunately, however, since the number of stars undergoing microlensing events at any given time will be small, relatively few stars actually need to be monitored with high cadences in order to detect possible planetary perturbations.

Traditionally, microlensing observations have been performed by groups using one of two distinct but complementary approaches. 'Search' or 'survey' groups use wide-field, low-cadence imaging to find new microlensing events; these groups then alert 'follow-up' groups who respond by coordinating an international network of telescopes to provide intensive, narrow-field coverage of a small number of active events.

Unfortunately, the relatively low cadences – partly due to small imaging fields of view, meaning many pointings were required to cover sufficient areas – of early microlensing

[4] There is a bias towards discovering distant lensing systems, since they have a greater probability of being observed along the same line of sight as a distant background source star.

survey efforts severely impeded the planet-detection prospects of follow-up groups. (For a detailed review of the history of microlensing campaigns, see Gaudi, 2012.) Over the years, much effort has been directed to improving survey cadences – so much so that today, the boundaries between survey and follow-up groups have started to blur, with most survey groups now spending at least part of their time engaging in very intense monitoring of specific fields.

Today, the two most prominent survey collaborations are the Microlensing Observations in Astrophysics group (Sumi et al., 2003) and the Optical Gravitational Lensing Experiment group (Udalski, 2003), both of which conduct microlensing surveys towards the Galactic Bulge. Both of these groups were involved in the discovery of the first microlensing-based discovery of an exoplanet (Bond et al., 2004), and they have been involved in all other microlensing-based exoplanet discoveries to date. MOA is currently in its second phase ('MOA-II') and carries out high-cadence (12 to 40 minutes) photometric surveys of ~ 50 million stars in Bulge fields, using a purpose-built 1.8-m telescope in New Zealand with a 2.2-deg^2 field of view. OGLE is currently in its fourth phase ('OGLE-IV'), and uses a 1.3 m-telescope in Chile with a 1.4-deg^2 field of view to observe hundreds of millions of Bulge stars, spread across multiple fields, using typical cadences of 20 to 60 minutes. OGLE-IV currently identifies about 1500 events per year, while MOA-II identifies about 600 events per year (Bond, 2012).

Current follow-up collaborations include the Probing Lensing Anomalies NETwork (PLANET; Albrow et al., 1998), RoboNet (Tsapras et al., 2009), the Microlensing Network for the Detection of Small Terrestrial Exoplanets (MiNDSTEp; Dominik et al., 2010), and the Microlensing Follow Up Network (microFUN, or μFUN: Gould et al., 2010). μFUN is a large, informal consortium – including many amateur astronomers, spread over five continents – which concentrates on observing high magnification microlensing events in detail.[5] The majority of the telescopes used by members are relatively small (0.25 to 0.4 m in diameter), though there are some 1.0- and 2.0-m class instruments involved as well.

From an observational perspective, the future of microlensing is bright. There has recently been an influx of new telescopes involved in microlensing, many more are scheduled to come online over the course of the next few years, and there are a number of exciting projects that have been proposed though await approval (Gaudi, 2012). For example, in 2014, the Korean Microlensing Network (KMTNet) is scheduled to start operating 1.6-m, 4-deg^2 field-of-view telescopes in South Africa, Chile and Australia, and will be able to provide uninterrupted, high-cadence (~ 10 minutes) monitoring of Bulge fields (Kim et al., 2010).

On an equally exciting note, microlensing observations might soon be carried out from space in the near future. The WFIRST (Wide Field Infrared Survey Telescope) – a proposed space-based telescope that will attempt to constrain the nature of dark energy alongside using microlensing to hunt for exoplanets – has been listed by the US

[5] Events with peak source amplification factors of $A \gtrsim 20$ are usually termed 'high magnification events'; since such events often reach peak magnitudes of $I \lesssim 15$, they can be monitored with small apertures. See Dong et al. (2009) for a remarkable example of a high-magnification event.

National Research Council as the top-priority mission for the next decade (Boss et al., 2011); ESA's recently-funded Euclid mission may also feature a microlensing component as part of its 'legacy science' programme (Beaulieu et al., 2010). Observing from space will provide many advantages: better photometry and uninterrupted coverage of active events will allow detections of analogues of all Solar System planets (masses $\gtrsim 0.1\ M_{\oplus}$, and orbital radii $\gtrsim 0.5$ AU) except for Mercury, including free-floating planets, and will also enable unique determinations of the masses of most extrasolar planets discovered by microlensing (Gaudi, 2012). As Bennett & Rhie (2002) noted, even the microlensing signals of exoplanetary *moons* should be detectable from a space-based microlensing observatory.

2.1.4 Modelling microlensing events

The general mathematical framework used to convert an observed microlensing lightcurve into useful information about the underlying physical system is that of the 'inverse problem'. Inverse problems are some of the most important and well-studied of all mathematical problems, and they arise in every conceivable branch of the physical, natural and even social sciences (Tarantola & Valette, 1982; Press et al., 2007).

The basic objective of a physical inverse problem is, given a parametric model that describes a physical process, and given observations of that process, to find the model parameters that leads to the best match between the observed data and the model predictions (Romanov, 1987). Inverse problems are so-named because they are considered to be the inverse of the (typically much simpler) 'forward problems' that entail mapping model parameters to predicted observations. The relationship between inverse and forward problems may be visualised thus:

$$\text{model parameters} \underset{\text{inverse problem}}{\overset{\text{forward problem}}{\rightleftarrows}} \text{data} \tag{2.1}$$

A simple example of a physical inverse problem would be trying to choose parameters for a Voigt-profile when performing a fit to a spectral line; once determined, the best-fitting parameters may be used to draw some conclusions about the physical processes underlying the spectral line. Similarly, when dealing with a microlensing lightcurve, one must reconcile a physical model with observed data – i.e. solve an inverse problem – in order to 'access' the physics underlying the observed lightcurve.

Some inverse problems, especially those involving linear models, are quite easy to solve, whilst many others are ill-posed and difficult to solve. (As discussed in the next chapter, difficult inverse problems are most often solved using the methods of mathematical optimisation.) Unfortunately, the inverse problems associated with microlensing modelling tend to be extremely difficult to solve (Vermaak, 2007).

Luckily, comprehensive and well-founded models do exist to describe a wide spectrum of possible microlensing scenarios, ranging from simple (a single star lensed by another star) to enormously complicated (multiple sources lensed by an arbitrary number of massive bodies): making life somewhat easier for microlensing modellers, the simplest

few of these models are sufficient to describe the vast majority of observed microlensing events.

Nevertheless, given the inevitable difficulty of the inverse problems associated with microlensing models, the interpretation of microlensing data tends to be time-consuming and fraught with difficulties, especially when working with any model more complicated than the most basic single-lens model.

The balance of this chapter focuses on two specific microlensing models (and the difficulties they pose to modellers), viz. a model that describes an isolated mass lensing a point-source (a model used to characterise the majority of observed microlensing events), and a model that describes a binary system lensing a point-source (a model useful for characterising most microlensing events involving an exoplanet). The latter model will play a central role throughout most of this dissertation.

2.2 The single-lens model

The simplest possible microlensing model describes a point-like source, lensed by a single, massive body, where the lens, source and observer are assumed to be in relative *rectilinear motion* (a static model would be even simpler, though in such a model the lensing effect would not vary as a function of time, so there would be no observable lightcurve), and it is such a model that we consider here. This basic model is sufficient to characterise the majority of observed microlensing events (involving, as they do, isolated stellar lenses and sources).

Assuming one is comfortable with the mathematics of general relativity, it is not too difficult to construct this model from scratch, starting only from the linearised Einstein field equations – which are valid for the relatively weak gravitational fields found e.g. at the exterior of stars – and the assumption of the Schwarzschild metric (Schwarzschild, 1916). The derivation draws on a number of geometrical insights and is, unfortunately, somewhat lengthy, so it will not be presented here (in any event, most standard texts on general relativity present the main details; see e.g. Misner et al., 1973, or Hobson et al., 2006); only the relevant equations governing the final model are presented below.

For a point-source, single-lens event, then, the amplification of the source's flux, as a function of time t, is given by Paczyński (1986):

$$A(t) = \frac{u(t)^2 + 2}{u(t)\sqrt{u(t)^2 + 4}} \geq 1, \tag{2.2}$$

where

$$u(t) = \sqrt{\left(\frac{t - t_{\rm m}}{t_{\rm E}}\right)^2 + b^2}. \tag{2.3}$$

Here, $u(t)$ represents the lens-source separation in units of the angular 'Einstein radius', denoted θ_E (see definition below); $t_{\rm m}$ is the time of maximum amplification, corresponding to closest lens-source approach; $t_{\rm E}$ is the time required for the source to transit the angular distance θ_E (without loss of generality, we assume that all relative motion

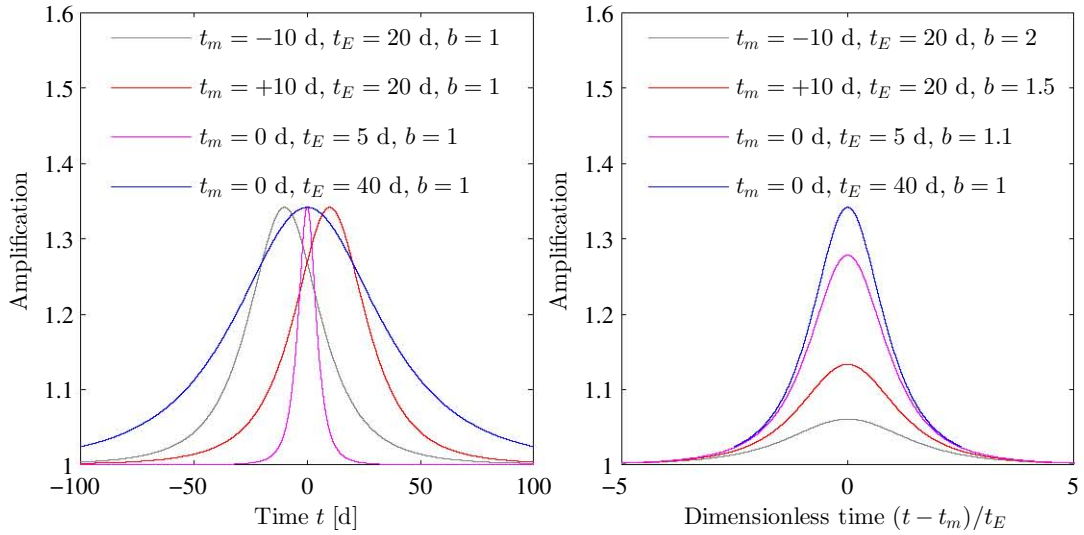

Figure 2.5 – Illustration of the effects of different values of the parameters $t_{\rm m}$ and $t_{\rm E}$ (left panel), and b (right panel) on single-lens lightcurves; note that the time axis has been rescaled in the right panel, to suppress the effects of different values of $t_{\rm m}$ and $t_{\rm E}$. Roughly speaking, $t_{\rm m}$ defines the time of the lightcurve's peak, $t_{\rm E}$ sets the lightcurve's width, and $1/b$ sets its maximum height.

is due to the source); and b is the minimum source-lens angular separation, or impact parameter, in units of θ_E.

The so-called angular Einstein radius[6] is related to the physical parameters of the lens system via:

$$\theta_E := \sqrt{\frac{4GM}{c^2}\frac{D_S - D_L}{D_S D_L}}, \tag{2.4}$$

where G is the gravitational constant, M is the mass of the lens, c is the vacuum speed of light, D_L is the distance between the observer and the lens, and D_S is the distance between the observer and the source. Apart from simplifying the lens equations, and setting angular scales for lensing events, θ_E has an important physical interpretation: the lensed image of a point source that is exactly aligned with a point lens will appear, to an observer, as a perfect ring, with radius θ_E, around the lens (Einstein, 1936). In the case of non-perfect alignment, the source will be split into two lensed *images*, one outside the Einstein ring, and one inside the Einstein ring (Kochanek et al., 2001); the observed amplification will be the linear sum of the amplifications of the individual, unresolved images. For Galactic Bulge microlensing events, with typical lens masses ($\sim 0.1 - 1\,M_\odot$) and source distances ($\sim 1 - 10$ kpc), $\theta_E \lesssim 1$ mas (Gaudi, 2010).

Typical timescales for events towards the Galactic Bulge will be $t_{\rm E} \sim 1$ month, but this can range from less than a day to many years (Bennett, 2008).

[6] It is interesting to note, for comparison, that the Schwarzschild radius of the lens is given by $(2GM)/c^2$ (Schwarzschild, 1916).

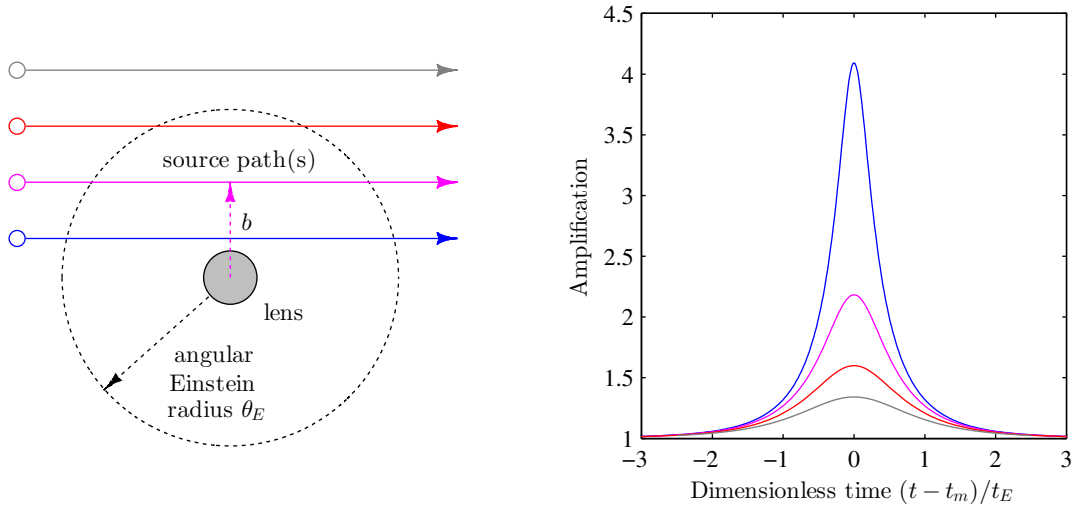

Figure 2.6 – Lensing geometry assumed in the simple, single-lens model (left panel), along with typical lightcurves for different possible source trajectories (right panel). The closer the source approaches the lens, as viewed by the observer (note the radial symmetry with respect to the lens), the greater the resultant magnification.

Finally, the amplification of the source needs to be related to some 'baseline' brightness or flux. Astronomers often favour the 'magnitude' measure of the (negative) logarithmic brightness of an object; if the source's unlensed magnitude is m_0, its lensed magnitude will be given by:

$$m(t) = -2.5 \cdot \log_{10} A(t) + m_0; \qquad (2.5)$$

this relation follows simply from the definition of apparent magnitude (Carroll & Ostlie, 2006). Note that $\forall\, A(t) > 1, m(t) < m_0$.

The single-lens events described by this model[7] are characterised by non-repeating, smooth, and symmetric lightcurves, rather reminiscent of Lorentzian functions; Figs. 2.5 and 2.6 refer.

This simple model turns out to be useful for describing a wide variety of 'ordinary' microlensing events (Schneider et al., 2006). However, despite the fact that the lightcurves of such events can, in principle, be described by an analytic function (Eqn. 2.2) that involves only a handful of parameters (t_m, t_E, b, and possibly also m_0, which may or may not be well-constrained from observations of the unlensed source), actually using the model is, in practice, rather less trivial than one might imagine. For example, the nonlinearity of the model leads to nontrivial parameter ambiguities and degeneracies, and this problem is compounded by the fact that photometric measurements of real lightcurves not only contain outliers, but also exhibit complicated, non-Gaussian statistics that vary between different observing sites (Dominik, 2008, 2009).

[7] The model's lightcurves are sometimes referred to as 'Paczyński curves', after Bohdan Paczyński (1940–2007), a Polish astrophysicist who made pioneering contributions to the field of microlensing.

2.3 The 'standard' binary-lens model

The model introduced in the previous section featured a point-like source lensed by a single, massive body. This model actually follows quite readily from a general, so-called 'lens equation' (see Bourassa, Kantowski & Norton, 1973; Paczyński, 1986), which describes the lensing effects due to an arbitrary number of point-like lenses (the lens equation itself may be derived from linearised relativity, along with various geometric considerations, in what is nothing more than a generalised version of the construction of the single-lens model). Indeed, a binary-lens model may also readily be 'extracted' from the general lens equation (Witt, 1990); unfortunately, however, the presence of two lenses, rather than just one, renders a binary-lens model *far* more mathematically complicated than the (already deceptively simple) single-lens model introduced in the previous section.

The binary-lens model introduced in this section (the 'standard' or simple binary-lens model, henceforth 'SBLM'), and used throughout this dissertation, features 8 basic parameters to describe rectilinear motion of a point-like source across a binary lens. The possibility of blended background light is allowed for, and the bodies in the lens are assumed to be static relative to each other. This model serves as a first-order means to characterise the majority of microlensing events involving an exoplanet: as shown by Gaudi, Naber & Sackett (1998), even a system with one star and multiple planetary bodies can often be well-approximated either by ignoring multiple planets, or by treating each planet plus its host star as an independent binary system, provided source magnification is not too high. (See Gaudi et al., 2008, however, for a rare example of an event for which a *triple-lens* model was required to give an adequate fit to the observed lightcurve.)

Again for brevity's sake, a detailed derivation of this model – whether from first principles, or from the aforementioned lens equation – will not be given here. Nevertheless, since binary-lens microlensing does take centre-stage in this dissertation, this binary-lens model will be developed and explored in rather more detail than the simpler model introduced in the previous section.

2.3.1 Parameterisation of lensing events

The SBLM features 8 parameters that characterise the basic geometry and dynamics of a binary-lens system and source. These parameters (and their physically-permissible ranges, where applicable) are as follows:

(i) m_0, the unlensed source magnitude, i.e. the magnitude of the source when it is not undergoing amplification by the lens;

(ii) $t_{\rm E}$, the Einstein radius crossing time, i.e. the time that it takes the source to travel an angular distance θ_E ($0 < t_{\rm E} < \infty$);

(iii) $t_{\rm m}$, the time of closest projected approach by the source to the primary lens;

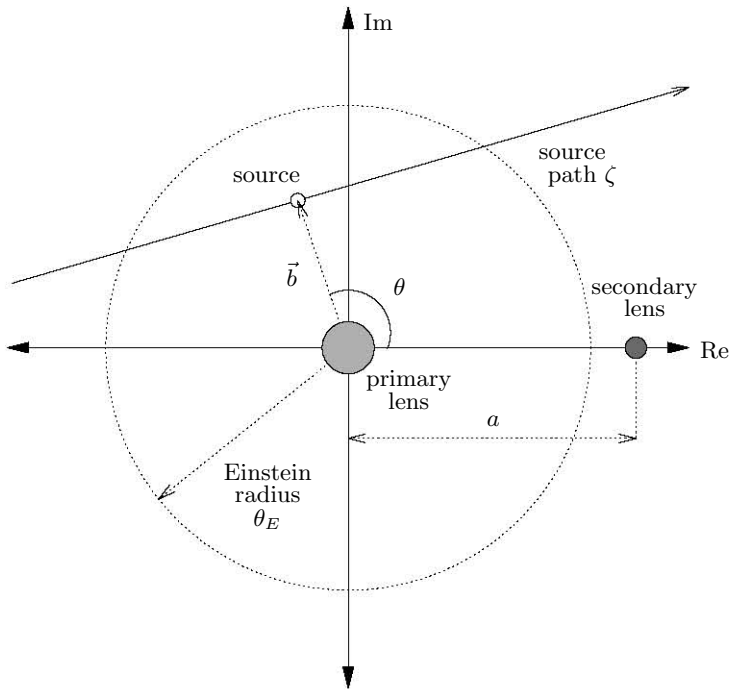

Figure 2.7 – Lensing geometry assumed in the 'standard' binary-lens model.

(iv) b, the length, in units of θ_E, of $\vec{b}$, where $\vec{b}$ is the projected impact parameter *vector*, i.e. the vector connecting the angular position of the source to the angular position of the primary lens, at the time of closest approach ($0 < b < \infty$);

(v) θ, the crossing angle, i.e. the angle formed between the axis connecting the lenses and the impact vector $\vec{b}$, as projected onto the sky ($0 \leq \theta < 2\pi$);

(vi) q, the ratio of the mass of the secondary lens to the mass of the primary lens ($0 < q < 1$);

(vii) a, the projected angular orbital separation, in units of θ_E, between the primary and secondary lens ($0 < a < \infty$); and

(viii) f, the blending parameter, i.e. the fraction of the observed baseline flux attributable to the *source* that ends up being lensed, as opposed to the baseline flux attributable to *other* luminous sources (e.g. stars) along the line of sight to the lens ($0 < f \leq 1$).

To obtain physical angular distances, the parameters a and b must be scaled by θ_E, where θ_E is the Einstein radius of the *primary* lens, defined as per Eqn. 2.4. (Of course, in the limiting case of an equal-mass binary, i.e. $q \to 1$, either lens could be assumed to be the 'primary'.)

For normal host stars (M-dwarfs with masses $\sim 0.3\ M_{\odot}$) towards the Galactic Bulge, a mass ratio of $q \sim 10^{-3}$ will correspond roughly to a Saturn-mass planet; and at distances of a few kiloparsecs from the Earth, where $\theta_E \sim 1$ mas, projected separation parameters of $a/\theta_E \sim 1$ will correspond to physical projected separations on the order of a few AU. For example, the best-fitting model for the 2.6 $M_{\rm J}$ planet discovered in 2003 by Bond et al. (see Sec. 2.1.4) had a mass-ratio parameter of $q = 0.004$, while the fitted parameter value $a/\theta_E = 1.12$ corresponded to a projected star-planet separation of 4.3 AU.

The first four SBLM parameters (m_0, $t_{\rm E}$, $t_{\rm m}$, and b) are direct analogues of the parameters used to describe single-lens events, while the next three (θ, q, a) are new parameters used to describe the basic properties of the binary system. The final parameter, f, is not binary-specific: blending is certainly possible in single-lens events, although this possibility was deliberately avoided in Sec. 2.2, if only to keep the explication of the single-lens model as straightforward as possible.[8]

The lensing geometry assumed by the SBLM is illustrated in Fig. 2.7.

For notational simplicity, the SBLM's 8 parameters can be arranged into a 'parameter vector', which we define via

$$\vec{p} = (p_1, p_2, \ldots, p_8) := (m_0, t_{\rm m}, t_{\rm E}, b, a, \theta, q, f), \tag{2.6}$$

so that each $\vec{p}$ specifies uniquely a single point in the 8-dimensional SBLM parameter space, and also (indirectly at least) an SBLM lightcurve. While many other parameterisations of binary microlensing events are possible (see e.g. Mao & Di Stefano, 1995; Dominik, 1999a), most are qualitatively similar to the one used by the SBLM.

2.3.2 Computing source amplification

Perhaps the biggest difficulty associated with the SBLM is that, unlike the single-lens model, it does not provide an analytical expression for the amplification of the source as a function of time and lensing parameters. Instead, as will be described below, the lens geometry, including the (time-varying) position of the source relative to the lens, must first be used to calculate the position of the *images* of the source that are formed by the lens; the light from each of these images is then summed to calculate the total amplification.

Firstly, note that whereas single-lens models are radially symmetric with respect to lenses, so that lightcurves will not be affected by the angle at which a source approaches a lens, this can no longer be the case with a general binary lens. Thus it becomes necessary to adopt a coordinate system to describe the orientation of the binary masses, as projected onto the sky, relative to the source (the introduction of the model parameter θ, or equivalently, the upgrading of b to a vector $\vec{b}$, reflects the same need to describe the orientation of the lensing system with respect to the source trajectory).

[8] In practice, many different observatories using many different filter bandpasses will contribute data to a microlensing event. Because source flux and blending will vary according to bandpass (and blending will also vary according to resolution), it is necessary to calibrate data between different observatories before performing fits using the SBLM formalism. Fortunately, this is a relatively straightforward task (Gaudi, 2010), and therefore such calibrations will not be considered in this work.

For convenience, complex notation is used here to describe the lensing system. The origin of the coordinate axes is placed at the projected position of the primary lens; with no loss of generality, the secondary lens is placed on the real axis; and $\zeta \in \mathbb{C}$ is used to denote the position of the source on the sky.

Straightforward geometric considerations, along with the assumption of rectilinear motion, yield the *source* position as a function of time, and some of the lensing parameters:

$$\begin{aligned}\zeta(t \mid b, \theta) &= t' \sin\theta + b\cos\theta + i\left(-t'\cos\theta + b\sin\theta\right) \\ &= (b - it')e^{i\theta},\end{aligned} \tag{2.7}$$

where t' is a dimensionless time parameter (specifically, t' measures time in units of $t_{\rm E}$, with the zero of t' corresponding to $t = t_{\rm m}$):

$$t' = \frac{t - t_{\rm m}}{t_{\rm E}}. \tag{2.8}$$

Assuming a binary lens and zero external gravitational shear (tangential stretching of source images around the lens, due to lensing; see e.g. Schneider et al., 2006), the general lens equation yields the following relation (Witt, 1990):

$$\zeta = z - \frac{1}{\bar{z}} + \frac{q}{a - \bar{z}}, \tag{2.9}$$

which, with ζ known, is an implicit expression for the positions, $z \in \mathbb{C}$, of the lensed *images* of the source. It may be shown that this special case of the lens equation has always either $n = 3$ or $n = 5$ solutions, corresponding to 3 or 5 lensed images.

The component of the total amplification associated with each individual image is given by the amount that that image is 'stretched' due to the lens; mathematically, this is given by the inverse of the Jacobian determinant of the mapping in Eqn. 2.9. Thus, once the position of the ith lensed image has been solved for, the amplification due to that image is calculated via

$$A_i = \frac{1}{|\det \mathcal{J}_i|} = \left|1 - \left|\frac{1}{{z_i}^2} + \frac{q}{(a - z_i)^2}\right|\right|^{-1}, \tag{2.10}$$

where $\mathcal{J}_i$ is the 2×2 Jacobian of Eqn. 2.9, for the ith image:

$$\mathcal{J}_i = \frac{\partial(\zeta, \bar{\zeta})}{\partial(z_i, \bar{z}_i)}. \tag{2.11}$$

Finally, the amplification due to the n individual (unresolved) images is summed, so that the total amplification of the source is given by:

$$A_S = \sum\nolimits_{i=1}^{n} A_i = \sum\nolimits_{i=1}^{n} \left|1 - \left|\frac{1}{{z_i}^2} + \frac{q}{(a - z_i)^2}\right|\right|^{-1}. \tag{2.12}$$

Unfortunately, however, the presence of blended, unlensed light will generally dilute the amplification signal of the source; if blended light is present, the observed amplification will, assuming constant blending, be (Di Stefano & Esin, 1995):

$$A(t) = f \cdot A_S(t) + (1 - f), \tag{2.13}$$

where $A_S(t)$ is the microlensing amplification (Eqn. 2.12), and the parameter f is implicitly defined to be the ratio of the unlensed *source* flux, to the total unlensed flux (source flux, plus flux of all luminous sources along the line-of-sight to the lens, including possible flux from the lens itself) in the telescope beam.

Finally, if the source's unlensed magnitude is m_0, its lensed magnitude will be given (as in the single-lens case) by:

$$m(t) = -2.5 \cdot \log_{10} A(t) + m_0. \tag{2.14}$$

The dependence of amplification on time, t, is made explicit here to emphasise that all of the preceding calculations – determining image positions and stretching, summing the amplifications due to individual images, etc. – apply to a single point in time, t, and thus culminate in just a single point on a lightcurve. In other words, given a set of values of many different values of t (observation times, say), and a single $\vec{p}$ (8 trial parameters to describe the lensing event), generating the corresponding lightcurve described by the SBLM, i.e. the set of points $m(t \,|\, \vec{p})$, requires working through Eqns. 2.7–2.14 for every single value of t!

SBLM lightcurves tend to exhibit far more complicated morphologies than their single-lens counterparts; this is discussed, along with examples, in Sec. 2.3.5.

2.3.3 Solving the lens equation

The most computationally-intensive step involved in computing the amplification of a point-source, due to a binary lens, is solving the lens equation, i.e. Eqn. 2.9. In practice it is convenient to convert the lens equation into a polynomial equation, and to solve this polynomial equation, rather than trying directly to find all the solutions to Eqn. 2.9.

Following some nontrivial algebraic manipulation, Eqn. 2.9 can be converted into a complex-valued, quintic polynomial equation[9] of the form

$$f(z \mid \zeta, a, q) = \sum_{j=0}^{5} c_j(\zeta, a, q) z^j = 0, \tag{2.15}$$

where the coefficients $c_j \in \mathbb{C}$ are given by (the author verified the expressions below

[9] This complex polynomial equation is equivalent to two coupled, real polynomial equations of the same order. These coupled equations can, in fact – following much further algebraic manipulation – be reduced to a single, real algebraic equation, also of the fifth order (Asada, 2002).

using both analytical and numerical testing):

$$c_0 = -\zeta a^2, \tag{2.16}$$

$$c_1 = a\zeta\left[2(1+q) + a(a-2\bar{\zeta})\right] - qa^2, \tag{2.17}$$

$$c_2 = \zeta\left[-(1+q)^2 + a\left(\bar{\zeta}(4-a\bar{\zeta}) + q(2\bar{\zeta}-a) - a(2-\bar{\zeta}a)\right)\right] + (1+q)(aq+\bar{\zeta}a^2) - 2\bar{\zeta}a^2q, \tag{2.18}$$

$$c_3 = \zeta\left[(1+q)(a-2\bar{\zeta}) + 2\bar{\zeta}a(\bar{\zeta}-a)\right] - a\left[\bar{\zeta}a(a-\bar{\zeta}) + 2\bar{\zeta} - aq\right], \tag{2.19}$$

$$c_4 = (a-\bar{\zeta})\left[\zeta(\zeta+2a) - q\right] + \bar{\zeta}, \tag{2.20}$$

$$c_5 = \bar{\zeta}(\bar{\zeta}-a). \tag{2.21}$$

The *fundamental theorem of algebra* tells us that every univariate polynomial with complex-valued coefficients will always have as many complex roots as its degree, if each root is counted up to its multiplicity (Fine & Rosenberger, 1997). This is a convenient property because whereas the number of solutions (at least three, but possibly five) to Eqn. 2.9 will not be known *a priori*, Eqn. 2.15 will always have exactly five solutions. Two of these five solutions may not be true solutions to the lens equation itself, but these spurious solutions may readily be eliminated by direct substitution into Eqn. 2.9.

However, how does one actually find all the roots of a quintic polynomial? On the one hand, finding the roots of linear, quadratic, cubic and even quartic polynomials is relatively straightforward, and there are well-known formulae that express exactly their roots in terms of their polynomial coefficients. On the other hand, however, the Abel-Ruffini theorem proves rigorously that there can be no general closed-form solution (i.e. an analytical expression using a finite number of elementary algebraic functions) for the roots of polynomials of degree five or higher (King, 1996; Jacobson, 2009). Nevertheless, the roots of quintic and higher-degree polynomials can be approximated, with arbitrary accuracy, using any one of a number of numerical techniques tailored for this purpose, for example the Jenkins-Traub algorithm, or Laguerre's algorithm (Ralston & Rabinowitz, 2001; Press et al., 2007).

Thus, computing SBLM lightcurves boils down to a straightforward but tedious and computationally-intensive exercise in numerical root-finding.

2.3.4 Difficulties associated with using the model

Given a set of parameters $\vec{p}$, it is a nontrivial but relatively straightforward task to obtain the lightcurve predicted by the SBLM. Unfortunately, however, the associated inverse problem – i.e., given a lightcurve, to use the SBLM to try to figure out the parameters underlying the lightcurve (see Eqn. 2.1) – is a notoriously difficult task, and it is of course the solution to this inverse problem, not the forward problem, that is of value insofar as the analysis and interpretation of real-world microlensing data is concerned.

In fact, two of the SBLM's parameters, m_0 and f, are easier to extract from lightcurves than the other parameters. Even if m_0 (which defines the baseline/unlensed flux of a

source) is not already constrained by archival data, reliable constraints can be placed on the parameter via simple geometric analyses of the low-amplification wings of a lightcurve (Vermaak, 2007). Remarkably, a least-squares estimate for the blending parameter, f, can be obtained directly via a simple, linear equation (this because, as per Eqn. 2.13, observed amplification is affected only linearly by the blending parameter) – this is certainly impossible for all the other SBLM parameters (Jaroszyński & Mao, 2001). However, given the bias inherent in the least-squares estimator, and given also the likely ambiguities between competing models, it is generally not a good idea to try to constrain f via a least-squares estimator.

Still, m_0 and (possibly) f notwithstanding, it is very difficult to extract SBLM parameters from a lightcurve. Some of the difficulties associated with the inverse problem include the following.

(i) *Computational expense.* Generating a single N-point ($N = 1000$, say) lightcurve requires the roots of N different quintic, complex-valued polynomials to be found; all the aforesaid roots also need to be checked as potential solutions to the lens equation (Eqn. 2.9). This alone cripples any method that relies on an exhaustive search through the parameter space.

(ii) *Volume of parameter space.* The SBLM has an 8-dimensional parameter space, and the physically-realistic ranges for some of the parameters (e.g. lens separations, lens mass ratios) can span many orders of magnitude. Vermaak (2003) estimated that for typical microlensing events, an exhaustive grid search of the SBLM parameter space would take on the order of 1000 years to complete (assuming a 2% error tolerance on all parameters, conservative bounds on possible parameter values, and an optimistic $\sim$ 2 ms/lightcurve calculation!). All extensions to the SBLM only exacerbate this problem.

(iii) *Extreme nonlinearity.* The mapping from SBLM parameter space to source amplification is highly nonlinear, and consequently, nearly-identical parameter sets can give rise to dramatically different lightcurves. A typical regression surface (e.g. χ^2-surface) will contain a large number of local optima, will be non-smooth, and will contain no clue as to where the global optima are to be found; furthermore, the wells of convergence around optima tend to be extremely small (Vermaak, 2003; Bennett, 2010). Thus, gradient-based fitting approaches, for example, will have little or no chance of successfully performing fits to observed lightcurves. To make matters worse, when dealing with noisy data, the true solution need not even correspond to a globally optimal solution (see Chapter 3).

(iv) *Parameter degeneracy.* The aforesaid problem of similar parameters leading to very different lightcurves can be overcome by making a denser sampling of the parameter space; more challenging, however, is the problem that a number of very *different* parameter sets can give rise to virtually indistinguishable lightcurves (Gaudi & Gould, 1997; Dominik, 1999a). Such degeneracies can be resolved by using data other than photometric (e.g. spectroscopic) to reduce the space of feasible parameters, or they can be avoided with high quality photometry and dense lightcurve

sampling (Mao & Di Stefano, 1995; Bennett, 2008). Failure to avoid or resolve degeneracies will lead to ambiguous model solutions, and therefore ambiguous conclusions about the physical characteristics of a lensing event.

The impasse, then, is that there is little hope of finding the 'correct' set of parameters to describe a binary-lens event if one does not make a dense and extensive search of the parameter space, but the computational burden of doing so is often crippling. A thorough search is especially important because even if one does happen to find one set of parameters that seems to provide a very good description of the event in question, there may exist many other parameter sets that provide equally good or better descriptions of the event. Any search or optimisation algorithm (see Chapter 3) used to model binary-lens events, then, must be able to locate – in a reasonable amount of time – *all* feasible solutions, i.e. all solutions compatible with the available observational data. Actually choosing between competing solutions/model parameter sets is a different problem altogether (chiefly a question of statistics, in fact; see e.g. Burnham & Anderson, 2002), and not one that will be addressed in this dissertation.

2.3.5 Morphologies of SBLM lightcurves

SBLM lightcurves exhibit a very wide variety of morphologies. Some lightcurves are practically indistinguishable from simple point-lens lightcurves, while others exhibit complicated structure including significant asymmetry, non-smoothness, and multiple rounded or kinked peaks (Mao & Paczyński, 1991; Night, Di Stefano & Schwamb, 2008). In fact, an SBLM lightcurve can contain anywhere between one and ten peaks (Liebig & Wambsganss, 2010), though lightcurves with more than four peaks are very rare. Monte Carlo simulations performed by the author, using physically-realistic parameter ranges for SBLM parameters, indicate that there is less than a 1% chance of a randomly-selected SBLM lightcurve containing more than four peaks – and even if a lightcurve does contain multiple peaks, there is a good chance that some of the peaks will be practically unobservable, on account of them having short durations, i.e. being very narrow, and/or on account of them having small amplitudes. SBLM lightcurves with single peaks are most common, double-peaked lightcurves are about half as common, and so on.

It is instructive to recall that source amplification under the SBLM is *inversely* proportional to the Jacobian determinant of the lens equation (which maps source positions to image positions); Eqns. 2.9 and 2.10 refer. From Eqn. 2.10, it is clear that there must be a family of image positions for which $\det \mathcal{J}_i$ will equal zero, implying the existence of solutions giving rise to infinite source amplification![10] The image positions satisfying $\det \mathcal{J}_i = 0$ form continuous curves referred to as 'critical curves', which are in turn mapped into curves in the source plane referred to as 'caustic curves' (or simply 'caustics').

As will be noted in the next section, infinite source amplification is non-physical: as it happens, the SBLM assumes point sources, whereas the *finite* size of real sources

[10] A geometric interpretation is that at these positions, the lens equation maps an (infinitesimally small) area element in the source plane to an infinitely large area in the image plane.

will smooth out the $\det \mathcal{J}_i = 0$ singularity, and will ensure that source amplification will remain finite (though generally still very large) on caustic curves. In any event, critical and caustic curves play a fundamental role in defining the morphology of binary lightcurves.

Whereas the caustic 'curve' in the single-lens case is actually just a single point in the source plane (corresponding to perfect source-lens alignment, i.e. $u = 0$ in Eqn. 2.2), binary-lenses give rise to multiple sets of closed caustics, each of which is composed of many concave segments called 'folds' that meet at points called 'cusps'. Folds and cusps are so named because the local lensing properties of sources close to these caustic features are equivalent to the generic fold and cusp mapping singularities that feature in mathematical catastrophe theory (Petters et al., 2001; Gaudi, 2010).

An important property of folds and cusps is that their *local* lensing properties are universal, regardless of the global properties of the lensing system. Thus, the number, positions, and magnifications of the critical images of sources formed near caustics have generic scaling behaviours that can be described using analytic approximations – this is a particularly useful property because it allows one to analyse and characterise the high-magnification parts of a lightcurve (associated with caustic crossings, cusp approaches, etc.) separately and independently from the global lens model used to describe the lightcurve (e.g. Albrow et al., 1999b; Dominik, 2004a,b).

There is a very large body of literature devoted to the study of binary lensing behaviour near folds and cusps: unfortunately, there is no scope to discuss the topic here. (And though occasional reference to caustic structures will be made throughout this dissertation, none of the later work is founded on caustic curve theory.) For the reader wishing to learn more about caustic curves, good references include Schneider et al. (1992), Zakharov (1995), Gaudi & Petters (2002a,b), and Han (2006).

Even without delving into the intricacies of catastrophe theory, however, we can get some sense of the wide spectrum of possible SBLM lightcurve morphologies simply by plotting a few lightcurves for different sets of SBLM parameters. In fact, there are only two SBLM parameters that define the number, shape, size and locations of caustic curves, and indeed, of amplification throughout the source plane: a, the binary separation parameter, and q, the binary mass-ratio parameter. Accordingly, Figs. 2.8–2.12 plot typical amplification patterns that result from five qualitatively-different combinations of a and q; many more combinations are, of course, possible. (Without studying the parameterisation of caustic curves, it is not straightforward to understand how the amplification patterns arise from different values of a and q; see e.g. Witt, 1993.)

With an amplification 'map' fixed by the parameters a and q, the parameters b (impact parameter) and θ (crossing angle) specify the actual path a source will take across the amplification map. Thus, the parameters a, b, q, and θ together determine the primary features of an observed lightcurve. Also in Figs. 2.8–2.12 are several lightcurves generated from source tracks overplotted on the amplification maps in the same figures. An arbitrary number of further lightcurves can, in principle, be 'constructed' simply by considering one-dimensional slices through the amplification pattern in question; note, in particular, how subtle changes to the impact parameter and/or crossing angle can lead to very different lightcurves.

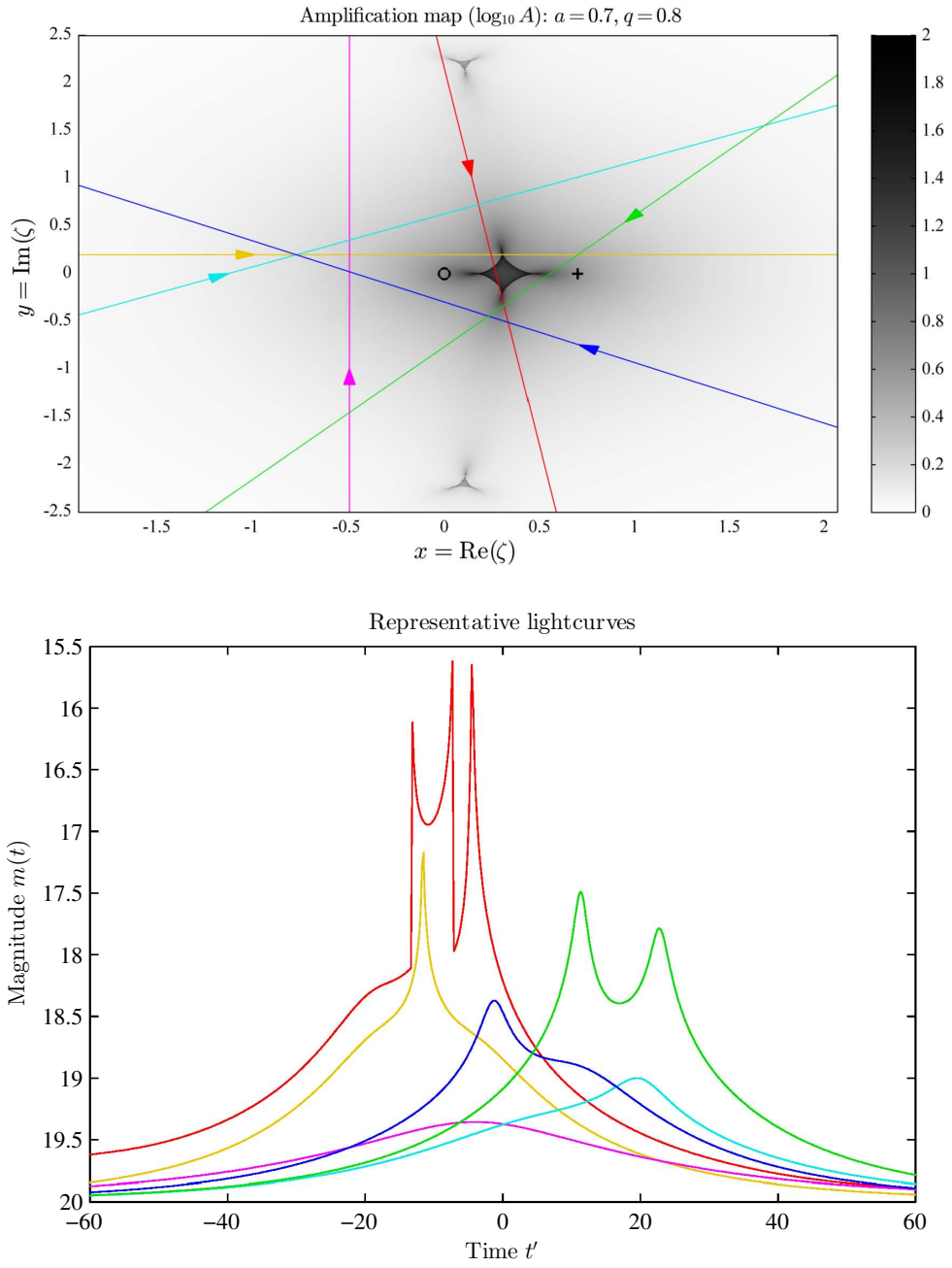

Figure 2.8 – Source-plane amplification map (upper panel) for a close-separation binary system ($a = 0.7$) with a relatively high-mass secondary lens ($q = 0.8$). The primary and secondary lenses are indicated, respectively, with a circle and cross in the source plane. The lightcurves in the lower panel correspond to the source tracks (defined by different values of b and θ) marked in the amplification map. Refer to the text in Sec. 2.3.5 for more details.

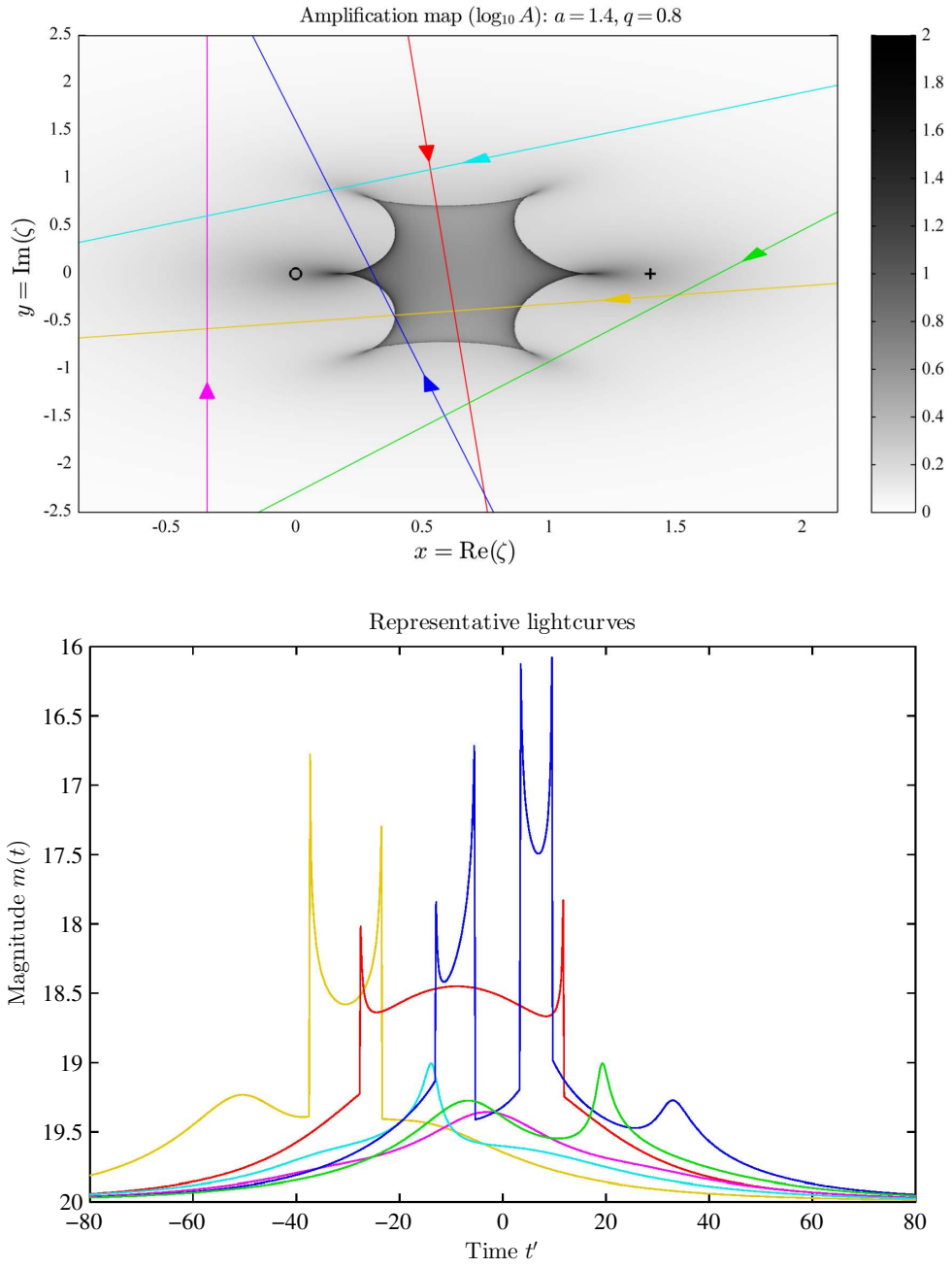

Figure 2.9 – Source-plane amplification map (upper panel) for an intermediate-separation binary system ($a = 1.4$) with a relatively high-mass secondary lens ($q = 0.8$). The primary and secondary lenses are indicated, respectively, with a circle and cross in the source plane. The lightcurves in the lower panel correspond to the source tracks (defined by different values of b and θ) marked in the amplification map. Refer to the text in Sec. 2.3.5 for more details.

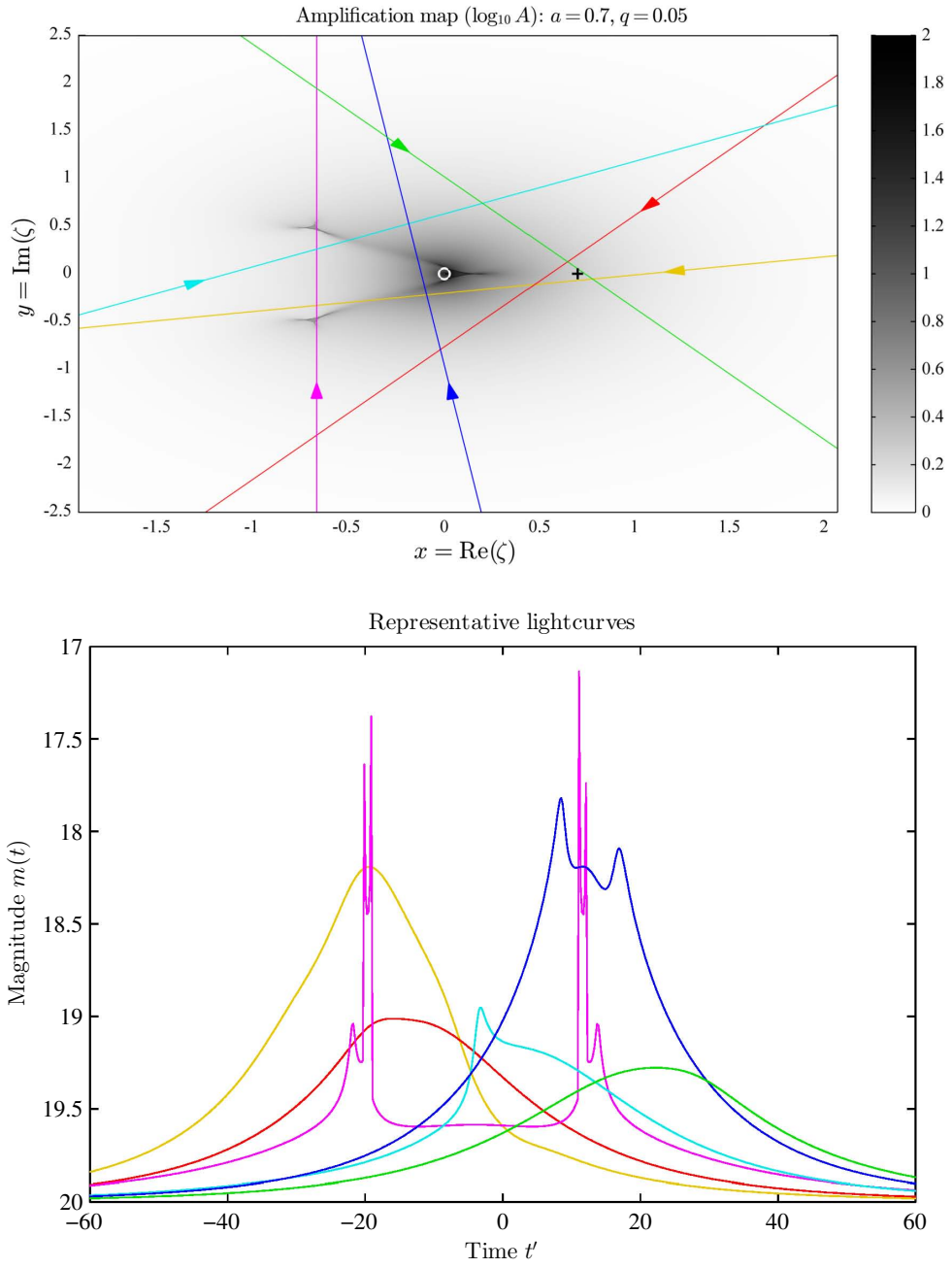

Figure 2.10 – Source-plane amplification map (upper panel) for a close-separation binary system ($a = 0.7$) with a relatively low-mass secondary lens ($q = 0.05$). The primary and secondary lenses are indicated, respectively, with a circle and cross in the source plane. The lightcurves in the lower panel correspond to the source tracks (defined by different values of b and θ) marked in the amplification map. Refer to the text in Sec. 2.3.5 for more details.

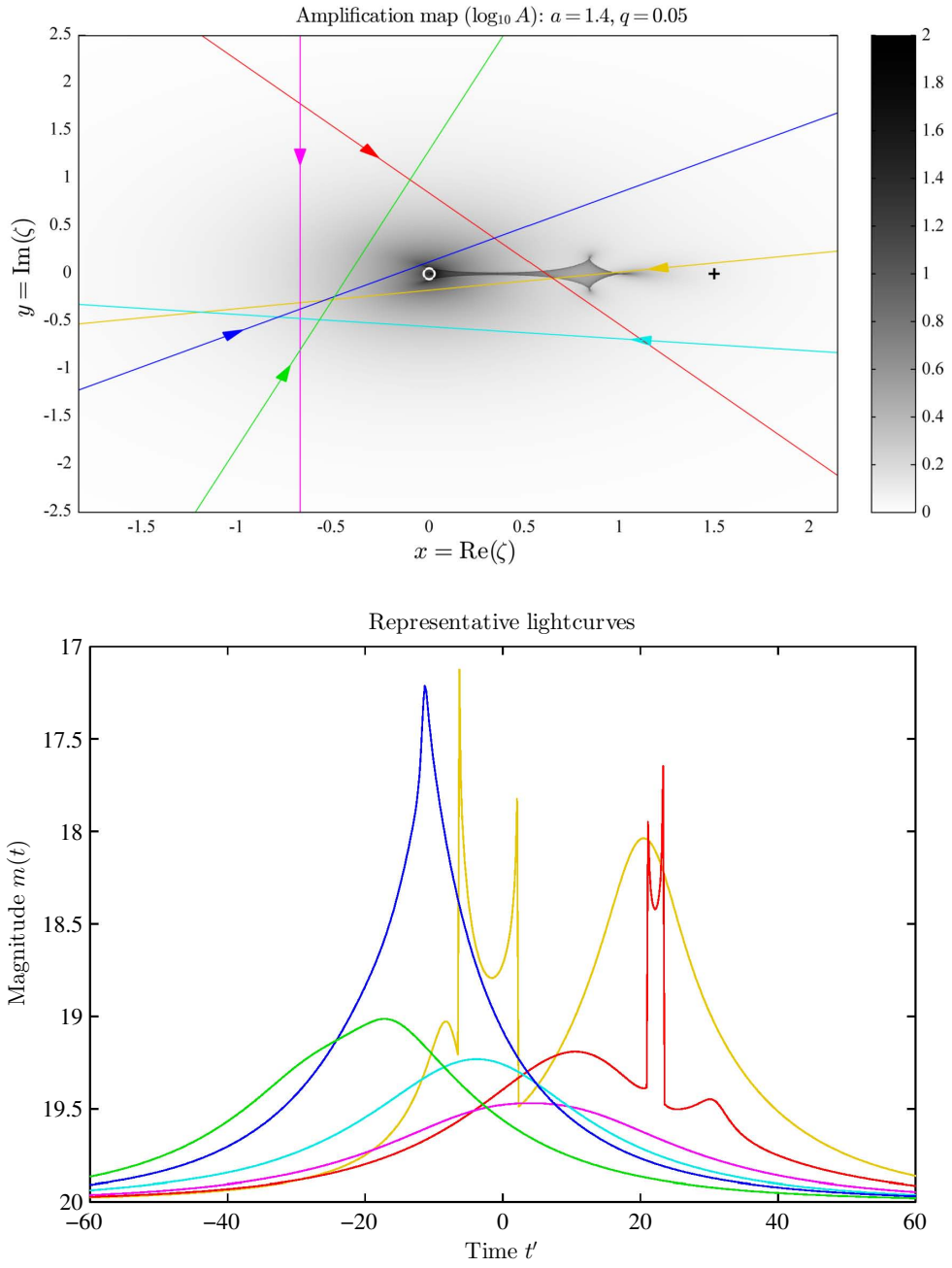

Figure 2.11 – Source-plane amplification map (upper panel) for an intermediate-separation binary system ($a = 1.4$) with a relatively low-mass secondary lens ($q = 0.05$). The primary and secondary lenses are indicated, respectively, with a circle and cross in the source plane. The lightcurves in the lower panel correspond to the source tracks (defined by different values of b and θ) marked in the amplification map. Refer to the text in Sec. 2.3.5 for more details.

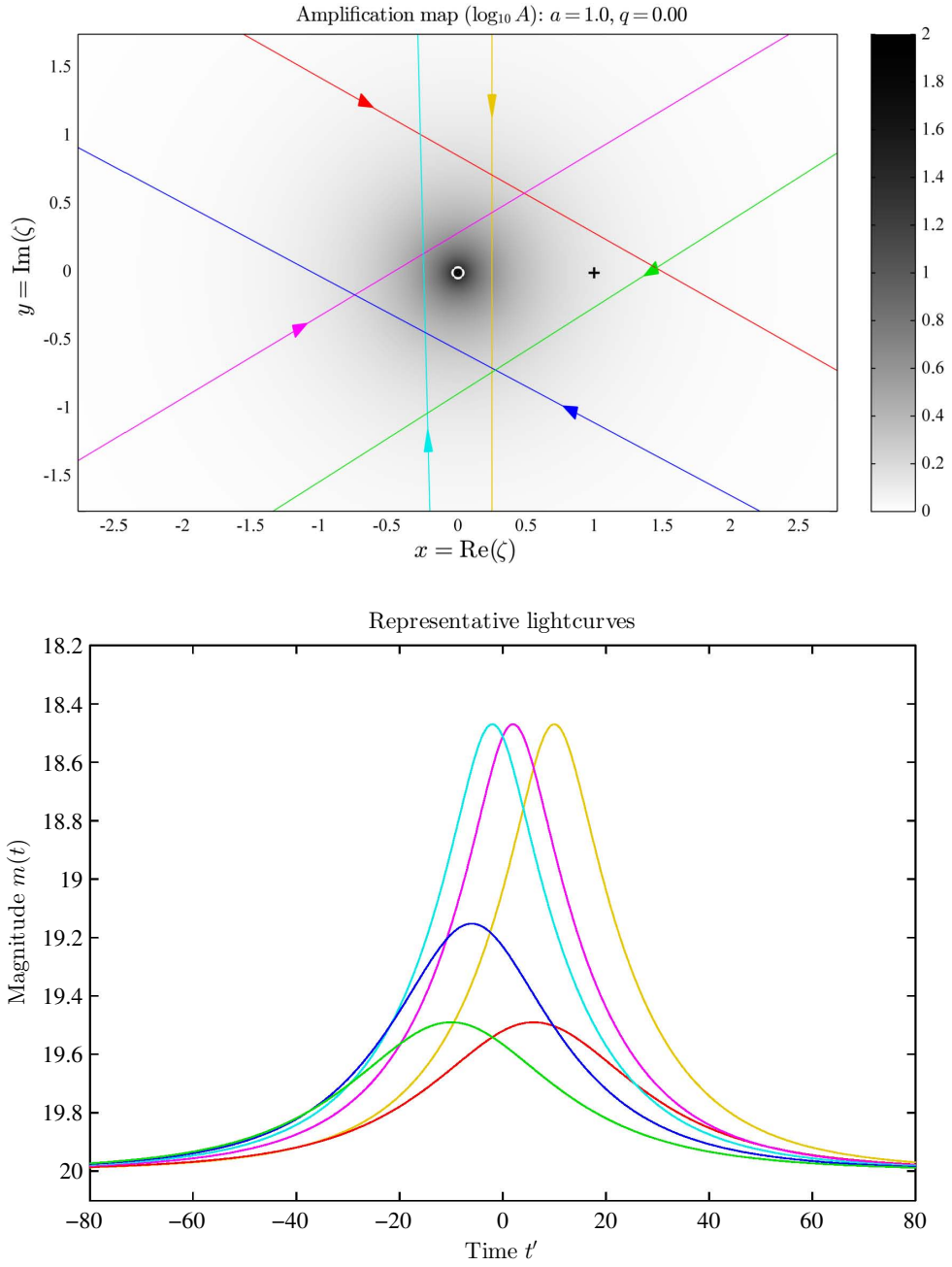

Figure 2.12 – Source-plane amplification map (upper panel) for the single-lens limit of a binary system, i.e. a binary with a zero-mass secondary ($q \to 0$; the infinite-separation, $a \to \infty$ limit is equivalent). The primary and secondary lenses are indicated, respectively, with a circle and cross in the source plane. The Paczyński lightcurves in the lower panel correspond to the source tracks (defined by different values of b and θ) marked in the amplification map. Refer to the text in Sec. 2.3.5 for more details.

The remaining SBLM parameters (m_0, t_m, t_E, f), all of which are carried over from the single-lens model, play only secondary roles in defining SBLM lightcurve morphologies, and in fact the role of each parameter may be understood qualitatively without any knowledge of the underlying caustic topology of an event. The parameter t_m applies a simple time-translation to a lightcurve; the parameter t_E sets the width of the entire lightcurve and all of its features (since it controls how long the source will take to track across the amplification map in question); m_0 sets the baseline flux of an event, and thus serves to translate the entire lightcurve up or down in magnitude space; and f, the blending parameter, serves to dilute microlensing amplification signals and thus 'flattens' all peaks. Scrutiny of the equations in Sec. 2.3.2 will confirm the claimed roles of these parameters. Given the secondary role of the parameters m_0, t_E, and f, these were all fixed to the same values for all the lightcurves in Figs. 2.8–2.11; the parameter t_m *was* varied, but only to minimise the degree to which plotted lightcurves obscured and overlapped with each other.

It should be noted that the lightcurves in the 'gallery' formed by the aforesaid figures were chosen simply to demonstrate the wide range of possible SBLM lightcurve morphologies, rather than to give a statistically representative picture of lightcurves that are likely to be observed in practice (for example, as noted earlier, binary lightcurves with more than a few peaks are relatively rare). For more comprehensive SBLM lightcurve galleries, refer to e.g. to Dominik (1999b), Kubas (2005) or Vermaak (2007).

2.4 Extending the standard binary model

The SBLM – which by itself is able to account for many and sometimes all of the complicated structures seen in real-world binary-lens lightcurves – is an unquestionably powerful model, and this model has proved quite successful in the interpretation of a number of binary microlensing events, including planetary events (e.g. Bond et al., 2004).

Nevertheless, this relatively simple model does neglect some of the second-order effects that often need to be taken into account when carrying out in-depth modelling of real binary-lens events (though even when such second-order effects are present, the model is often still able to provide first-order fits to lightcurves – see Chapter 6).[11] Fortunately, the model can be extended to take into account the second-order effects. This section provides a very brief look at the most significant second-order effects, their observable signatures, and their impact on fitting/modelling.

2.4.1 Finite source effects

At the moment that a point-source passes exactly over a caustic curve, the SBLM predicts that the source's amplification will be infinite. This is, of course, physically impos-

[11] It should be emphasised that this work does not attempt to advocate the SBLM for use in modelling real microlensing events: the model is adopted primarily to provide a relatively straightforward platform for benchmarking different algorithms studied later in this work (Chapter 5), and for answering various questions related to the modelling of ongoing microlensing events (Chapter 6). The shortcomings of the model will be flagged on numerous occasions later in this dissertation.

sible – amongst other physical laws, infinite amplification would violate the conservation of energy – which indicates that the SBLM is an imperfect model.

To be sure, it is not possible for the time of any real observation time to coincide exactly with the infinitesimal instants of infinite amplification predicted by the SBLM, so one generally need not worry about SBLM calculations actually 'blowing up' and spitting out infinities. However, especially with modern, high-cadence surveys, observations will often bracket very tightly the times of caustic crossings, in which case the amplifications predicted by the SBLM will nevertheless be far (albeit not infinitely) higher than the amplifications observed in reality.

This problem may be traced to the failure of the point-source approximation made by the SBLM. A resolved stellar source actually needs to be treated as a finite *disc* of light – often with a varying intensity across the disc, due e.g. to limb-darkening – and different parts of the disc will be amplified by different factors, based on their slightly different positions within the lensing field. Total amplification of the source thus needs to be computed as the integral of the amplifications of the infinitely many points that comprise the extended disc.

During a caustic crossing, a subset of the points comprising an extended source will still lie on a one-dimensional caustic of infinite amplification – however, their infinite amplification will be offset by their infinitesimally small combined area, and so the integral to compute the total amplification will remain well-behaved. To put this more precisely, the singularity occurring in Eqn. 2.10 when $\mathcal{J} = 0$ is *integrable*, so finite sources will always have finite amplifications (Bennett & Rhie, 1996).

Kubas (2005) showed that when one is dealing with giant stellar sources (radii $\gtrsim$ 10 $R_\odot$), the point-source approximation will generally be liable to break down; in a similar though more general vein, Vermaak (2007) suggested that it becomes important to model finite-source effects whenever sources have radii larger than about $0.01\theta_E$ (in other words, finite-source effects will be more significant for lower-mass lenses). In such situations, then, it is necessary to extend the SBLM by including a parameter to quantify the radius, ρ_*, of the source. The regions of peak amplification in lightcurves – e.g., regions associated with caustic crossings – are usually the only parts of lightcurves affected by finite-source effects. The effect of $\rho_* > 0$ on SBLM lightcurves is usually to smooth peaks out, making them broader and less pronounced, as one would expect from a dilution of the caustic effect.[12]

Vermaak (2000) showed that planet detection probabilities generally decrease with increasing ρ_*. On the positive side, however, careful analysis of finite-source effects in a well-sampled lightcurve sometimes allows one to probe the one-dimensional surface brightness profiles of sources, making microlensing one of the few successful techniques for measuring brightness profiles of stars (Albrow et al., 1999a), including star spots (Hendry et al., 2002), and the only one known to work for fairly distant stars, i.e. stars at a distance of several kpc.

[12] It is also possible for finite-source effects to become important during e.g. a cusp approach, where a point-source might only be moderately amplified, whereas a resolved source on the same trajectory might extend over the actual cusp, and thus have its limb amplified by a *larger* factor.

Although finite-source effects will usually affect only quite localised parts of a given lightcurve, these effects do show up in a large fraction of binary-lens events (Dominik, 2008) – and unfortunately, it tends to be enormously expensive to compute the amplification of an extended source. Extant approaches include direct numerical integration over source surfaces (Schneider et al., 2006), indirect numerical integration over the surfaces of lensed *images* (Bennett & Rhie, 1996), using Green's theorem to transform surface integrals over images into line integrals around their borders (Dominik, 1998a; Bozza, 2010), using semi-analytical approximations to calculate amplifications (Gaudi & Petters, 2002a; Gould, 2008), and more.

It is not within the scope of this dissertation to discuss the details of the aforementioned computational techniques; indeed, one could devote an entire dissertation to a single technique used to compute the amplifications of finite sources. The point relevant to this dissertation, however, is that having to deal with finite-source effects generally increases the computational cost of generating/fitting lightcurves by about two orders of magnitude (Vermaak, 2007). Therefore, when possible, it is obviously desirable to be able to model an event *without* having to take into account finite-source effects (even if only to generate first-order models; see Chapter 6). Given the prevalence of finite-source effects, however, dealing with them is very often a 'necessary evil' (Witt & Mao, 1994).

2.4.2 Parallax and xallarap effects

The change of an observer's position, induced by the orbital motion of the Earth around the sun, causes the observed motion of a source to deviate from the SBLM's assumption of rectilinearity. This leads to long-term deviations in lightcurves (Gould, 1992), in what is known as the 'orbital parallax' (a.k.a. 'annual parallax') effect.[13] Modelling this parallax effect requires the inclusion of two extra parameters, viz. parameters to describe the two components of the lens parallax vector, projected on the sky, in the north and east equatorial coordinates, respectively.

Analogously to parallax, if a *source* (as opposed to observer) undergoes significant acceleration over the course of a microlensing event – due to a binary companion, for example – the source's trajectory will be affected, again leading to a lightcurve that deviates from a canonical SBLM lightcurve. This is known as the 'xallarap' ('parallax' spelt backwards) effect (Dominik, 1998b). Assuming a circular orbit, and a very faint binary companion, the xallarap effect can be described by five additional parameters, viz. parameters to describe the orbital period, the inclination, and the phase angle of the source, and the two components of the xallarap vector in the north and east directions (Shin et al., 2011).

Although the actual calculations required to account for parallax/xallarap effects are

[13] Annual parallax is not the only possible type of parallax effect. For example, for sources very close to a caustic, small differences in perspective lead to slightly different amplifications being observed instantaneously by observers located at different observatories on the Earth, in an effect known as 'terrestrial parallax'; if observations are made in space, e.g. from a satellite in solar orbit, the effect will be more pronounced, and is referred to as 'satellite parallax' (Gaudi, 2010). Additionally, so-called 'diurnal parallax' results from the rotation of the Earth around its axis (An et al., 2002).

relatively straightforward – especially compared to those associated with finite source effects – having to deal with these effects does lead to an increase in the dimensionality and volume of the SBLM's parameter space, making modelling/fitting more difficult. Fortunately, parallax anomalies tend to occur over relatively long timescales, and so can usually be ignored when modelling e.g. planetary microlensing events, which tend to characterised by short timescales (Griest & Safizadeh, 1998). On the positive side, however, when parallax effects *are* measured, they can lead to direct constraints on physical parameters such as lens masses, distances, and projected separations (Gould, 1992).

2.4.3 Lens orbital motion effects

The assumed binarity of a lens implies that the positions of the lens components should vary over time, on account of their mutual orbital motions. This 'orbital' effect causes a change not only to the source position with respect to the lenses, but also to the lensing system's magnification patterns, since the projected binary separation will change over time (Dominik, 1998b). A number of different parameterisations of lens orbital effects are possible: for example, if the lens components are assumed to rotate with constant angular speed, two extra parameters are required, whereas if full Keplerian motion is allowed for, the SBLM needs to be supplemented with four new parameters (Skowron et al., 2011).

As with parallax effects, orbital motion effects tend to be difficult to detect, and can very often be ignored, especially when modelling planetary events, where deviations tend occur over relatively short timescales (on the order of days or hours), resulting in very narrow windows during which lens orbital motion (which, for planets, is characterised by timescales of years) can have an impact on lightcurves. In order to resolve lens orbital motion during planetary events, it is usually necessary to have a relatively complicated planetary deviation, with more than a single caustic crossing/cusp passage being very well sampled by the data (Bennett, 2008). However, if lens orbital effects *are* measured, they can allow one to constrain the lens system's orbital parameters, and the intrinsic separations between the lens components (e.g. Gaudi et al., 2008; Dong et al., 2009; Skowron et al., 2011).

2.4.4 Other effects, and beyond binary models

The literature contains discussions of a handful of additional, higher-order effects which could, in principle, affect binary-lens events – for example, effects associated with the finite physical sizes of *lenses* (Agol, 2002). In most cases, though, these higher-order effects are expected to unobservable, and/or extremely rare (Gaudi, 2010).

It should be noted that when modelling a binary-lens event, the microlensing effects induced by the presence of the secondary lens may be thought of as perturbations to the effects of the primary lens (Bozza, 1999). In other words, binary-lens lightcurves may be thought of as canonical, 'Paczyński' lightcurves (i.e. single-lens lightcurves), but with the addition of higher-order perturbations, due to the presence of the secondary lens.

Similarly, if more than two lenses are present, the effects of the additional lenses may be thought of as perturbations to the gross effects of the binary masses.

To be fair, however, the task of modelling even a triple-lens lightcurve is so enormously complicated that it would be a little absurd think of the addition of a third lens to a binary system as being a mere perturbative effect (Ryu et al., 2011). For example, modelling a triple-lens system requires the lens equation to be 'upgraded' to a tenth-order complex-valued polynomial, and the SBLM to be supplemented with three extra parameters (e.g. to describe the mass ratio, the projected separation, and the angle between the tertiary lens and the primary; other parameterisations are, however, possible). More generally, an N_l-lens version of the SBLM (neglecting finite source effects, parallax effects etc.) would require $3N_l + 2$ parameters, and the associated lens equation would be equivalent to a complex-valued polynomial of degree $N_l^2 + 1$ (Gaudi et al., 1998). Though such models would share some basic 'ingredients' with the SBLM, one might just as well regard them as being separate models altogether.

2.5 Conclusions

This chapter presented a brief overview of gravitational microlensing, and, in particular, of the microlensing model (the simple binary-lens model, or SBLM) that is to take centre stage throughout this dissertation.

It was noted that whilst gravitational microlensing is an important technique for detecting and characterising exoplanets, it tends to be extremely difficult to model and interpret the lightcurves associated with microlensing events. Even when modellers are able to employ the simplest possible model (viz. the SBLM) for describing the lensing action of an exoplanet and its host star, they are confronted with such nontrivial problems as enormous parameter spaces, extreme nonlinearity, ambiguous solutions, and more.

In spite of the development, in recent years, of many novel and ingenious techniques, modelling binary-lens events remains a challenging task; the lack of a demonstrably superior approach to modelling lightcurves is evidenced by the fact that the microlensing community has yet to converge on a single algorithm or technique (see e.g. Dominik, 2008; Kains et al., 2009; Mao, 2012). Accordingly, it remains important to develop new and improved techniques for modelling microlensing lightcurves.

It is worth mentioning here that – largely on account of the limited space in this dissertation – this chapter has been able to provide, at best, a broad-brush picture of microlensing, with emphasis on just a few topics germane to the balance of this dissertation; there are many important topics (including planet detection probabilities, difference imaging, pixel microlensing, etc.) that were not even touched upon in this cursory discussion. Fortunately, there exist several reviews that do provide far more comprehensive overviews of the field of microlensing: see, for example, Gaudi (2010) or Bennett (2008). The former review emphasises the theory and phenomenology of exoplanetary microlensing, while the latter gives a detailed discussion of the need for space-based microlensing missions. Other excellent reviews of microlensing are provided by Paczyński (1996), Gould (2005), and Mao (2008). The book by Schneider, Ehlers

& Falco (1992) is a trusted reference for all things related to gravitational lensing in general, while the book by Schneider, Kochanek & Wambsganss (2006) provides a more up-to-date discussion of many topics also drawn from across the different lensing regimes (strong lensing, weak lensing, and microlensing).

The next chapter of this dissertation considers techniques for solving a broad class of problems that subsumes those associated with microlensing modelling.

3
Evolutionary algorithms

This chapter provides an introduction to evolutionary algorithms, a class of versatile and robust search/optimisation algorithms that draw inspiration from evolutionary biology.

First, the general concept of nonlinear optimisation, and its connection to microlensing modelling, is introduced; evolutionary algorithms are then presented as a means for solving difficult nonlinear optimisation problems. Next, the chapter discusses the basic ideas underpinning evolutionary algorithms, outlines some of their advantages and disadvantages (with particular emphasis on their suitability for microlensing modelling), and finally gives examples of the many applications that these algorithms have already found in the fields of astronomy and astrophysics.

3.1 Introduction: nonlinear global optimisation

It is no exaggeration to say that optimisation – the concept that most students of elementary calculus are taught to associate, somewhat naïvely, with the idea of 'taking a derivative and setting it equal to zero' – lies at the very foundations of the physical sciences.

Indeed, a very large class of interesting mathematical problems can be (re)formulated as (global) optimisation problems. The solution of systems of algebraic or even differential equations, for example, can be cast quite naturally in terms of optimisation (Pardalos & Rosen, 1987). This holds true also for the all-important inverse problems that are ubiquitous in the physical sciences, i.e. problems where one seeks to transform experimental data into model parameters in order infer properties of the physical systems being studied. The problem of finding a physical model compatible with a microlensing lightcurve is, as noted in the previous chapter, nothing more than a (very difficult) inverse and thus optimisation problem.

The goal of a *global* optimisation problem is, given a so-called 'cost function'[1] $f : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$, to try to find a point (more generally, a set of points) $\vec{x}^* \in \Omega$ such that:

$$\forall \vec{x} \in \Omega : f(\vec{x}) \geqslant f(\vec{x}^*). \tag{3.1}$$

[1] Depending on the context, cost functions are also referred to as 'energy functions' or 'objective functions'.

The set Ω is referred to as the 'search space' or the 'choice set', the elements of Ω are called 'candidate solutions', and $f(\vec{x}^*)$ is called the global minimum.[2] It should be noted that the terms 'optimisation' and 'search' are often be used interchangeably, since any optimisation problem can generally be interpreted as a *search* for one or more good solutions in a space of candidate solutions.

The existence of a global minimum will depend on the nature of f and Ω. For example, if f is continuous and Ω is compact, then the Bolzano-Weierstrass theorem guarantees that f will attain its global minimum and maximum at least once (Stewart, 2008).

A local minimum, $f(\hat{\vec{x}})$, is defined by the condition.

$$\forall \vec{x} \in \Omega, \exists \delta > 0 : \left\| \vec{x} - \hat{\vec{x}} \right\| < \delta \Rightarrow f(\vec{x}) \geqslant f(\hat{\vec{x}}). \tag{3.2}$$

A simple optimisation problem is illustrated in Fig. 3.1.

Whereas finding an arbitrary *local* minimum of f will usually be a relatively straightforward task, especially if one has a good 'first guess' – extremely efficient techniques, e.g. hill-climbing methods, exist to solve such local optimisation problems – finding its global minimum/minima is a far more challenging problem. Furthermore, whilst very elegant

[2] Maximisation of $f(\vec{x})$ is, of course, equivalent to minimisation of $g(\vec{x}) := -f(\vec{x})$. Functions such as $g(\vec{x})$, which are to be maximised, are usually termed 'fitness functions'.

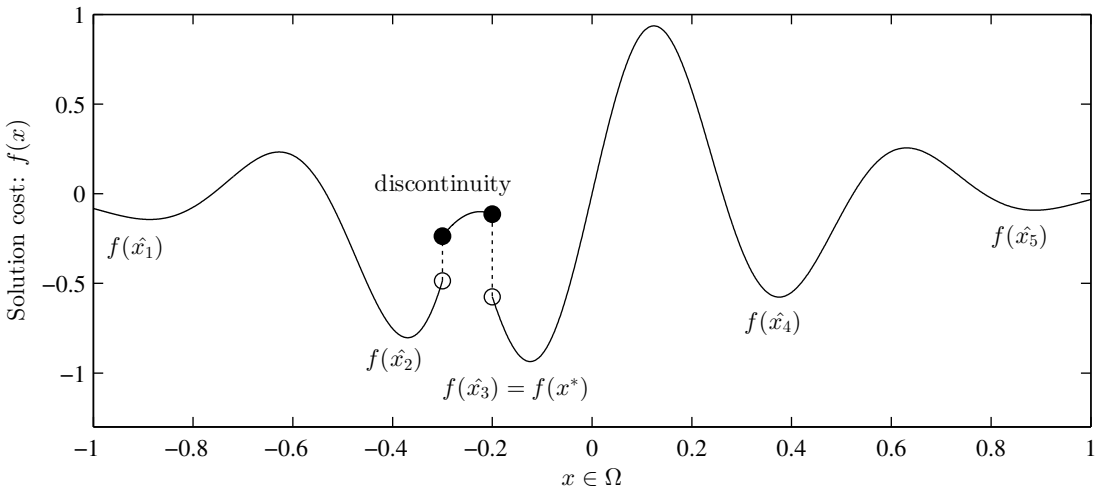

Figure 3.1 – A 'toy' optimisation problem where the cost function, $f(x)$, is a function of only one variable, and the search space is the set $\Omega = [-1, 1]$. The cost function features five local minima, only one of which corresponds to the global minimum, x^*, and two jump discontinuities. The goal of the associated optimisation problem would be to find the value of x that minimises $f(x)$ – i.e. to find x^* – a process that could correspond, say, to choosing a parameter in a simple physical model in order to minimise a fitting error. The 'correct answer' is $x^* = -1/8$, as the above plot suggests; of course, one would typically need to compute many values of $f(x)$ before one could arrive at such a conclusion (or plot the function in the first place).

and powerful mathematical apparatus – including the methods of linear programming – may be employed in the specialised case where f depends linearly on $\vec{x}$, in the more general case, this dependence will be nonlinear, and tractable analytical methods exist only for a very small subset of such *nonlinear* optimisation problems (Michalewicz & Fogel, 2000).

Alas, there is no general or foolproof approach to locating the global minima of nonlinear functions. Most established approaches – whether analytical, stochastic or (meta)heuristic[3] – to solving nonlinear global optimisation problems yield excellent results on some limited class of problems, but have drawbacks that cripple them when faced with certain (reasonably) difficult problems. For example, they might get stuck too easily in local minima, they might require functions to be analytic, they might be thwarted by discontinuities, or they might be too slow to be of practical value when faced with enormous search spaces (Charbonneau, 2002a).

As luck would have it, real-world cost functions tend very often to be nonlinear, non-analytic, multimodal, and/or discontinuous. Suffice it to say, then, that it is essential to look beyond simplistic, conventional algorithms if one is to have any hope of success when confronted with such challenges.

3.2 Evolution and evolutionary algorithms

The primary mechanism driving biological evolution is widely accepted to be 'natural selection', the phenomenon whereby 'fitter' individuals (individuals better adapted to their environment, be it in terms of finding food, avoiding becoming food, or competing for reproductive partners) will produce, on average, more offspring than their less fit competitors. This is because in any ecosystem, many members of a given species tend to die from predation or attrition before they have a chance to reproduce (Darwin, 1859). The individuals who *do* survive to reproduce will tend to possess traits that increased their chances of survival in the first place: hence, 'survival of the fittest'.

However, there are two additional ingredients required for natural selection to lead to evolution:

(i) inheritance: offspring must retain some of the features that made their parents fitter than average, otherwise evolution will be 'reset' after each generation; and

(ii) variability: individuals of varying fitness must exist within the population, otherwise natural selection will have nothing on which to operate, and evolution will cease.

Although these additional requirements were clear to Darwin and his contemporaries, their underlying mechanisms remained unexplained until the early part of the twentieth century; today, the processes through which heredity is mediated, and variability maintained, are well understood. In short, the information determining the physical makeup,

[3] A heuristic optimisation algorithm tries to make iterative improvements to candidate solutions, with respect to a given measure of quality, without making any restrictive assumptions about the function itself. A metaheuristic algorithm is one that combines different heuristics into a single algorithm.

growth and development of individuals is encoded as a linear sequence of genes, each of which can each assume a finite set of values. When two individuals reproduce, complementary portions of their genes are passed on to their offspring and combined to define that offspring's full genetic makeup. Thus heredity is mediated. However, truly random alterations ('mutations') occur occasionally to the values of the parents' genes, and copying mistakes also take place when these genes are passed on to offspring. These random changes, combined with the random splicing and complementary recombination of parental genes that occurs during the formation of offspring, ensures variability (Fisher, 1930).

The living individual can be thought of – to within a good approximation – as an outer manifestation of its defining genes, and it follows that an individual's fitness within a population may be thought of as a function of the values assumed by its genes. Evolution, then, is a process that drives an increase, over the course of many generations, in the average fitness of a population: the link to optimisation (formulated in terms of maximisation) should be clear.

Far from being a clumsy process that works only sporadically, time and time again evolution has produced feats of incredible natural engineering that, to this day, remain the envy of scientists at the very forefront of fields such as artificial intelligence, biology, engineering, and robotics. It is particularly notable that evolution manages to achieve its astonishing feats through processes that are known to be simple, self-contained and markedly stochastic (Mayr, 1963). One has to wonder, then, whether the processes underlying biological evolution could be usefully employed in the field of optimisation.

Enter evolutionary algorithms (hereafter EAs): a broad class of metaheuristic optimisation algorithms, inspired by biological evolution, that tend to yield good results on a very wide range of (even extremely difficult) optimisation problems. EAs incorporate, in a computational setting, notions such as natural selection/survival of the fittest, reproduction, genetic recombination, inheritance, and mutation. Many EAs also embody the notion of *adaptability* by incorporating mechanisms that allow autonomous, real-time changes to the actual algorithms, in response to the optimisation problems with which they are confronted.

The first EA-based mathematical optimiser was proposed in the mid-1970s (Holland, 1975), and since then, many modifications and improvements to the basic algorithms have been developed, including mechanisms without any direct biological analogues (Storn & Price, 1997; Haupt & Haupt, 2004).

3.3 A typical EA: the genetic algorithm

So-called 'genetic algorithms' (GAs) form one of the most successful subsets, and certainly the best-known subset, of evolutionary algorithms; 'evolution strategies' (ESs), developed independently from (though more or less concurrently with) GAs, form the next best-known subset (Bäck, 1996). Two final subsets worth acknowledging, though there is no scope to discuss them in this dissertation, are 'differential evolution' (Storn & Price, 1997) and 'genetic programming' (Koza & Poli, 2005) algorithms.

In any event, in spite of the rich variety of their potential incarnations, most EAs do share a basic working scheme. To illustrate the workings of a typical EA, then, let us consider the canonical genetic algorithm.

The genetic algorithm starts with a large, randomly-generated population of candidate solutions (called individuals, or phenotypes) to the optimisation problem at hand, and associates with each solution an encoded version of the phenotype (called a chromosome, genotype, or an individual's genetic material), as well as a problem-specific measure of the solution's quality (fitness). Then, by repeated application of 'genetic operators' mainly at the *genotypic* level, the algorithm causes the population as a whole to increase in phenotypic fitness: that is, solutions are made to evolve towards optimality.

A typical (though simplistic, and by no means general or optimal) working scheme for the GA is as follows:

(i) construct a random initial population of genotypes, represented by binary strings;

(ii) decode the genotypes, and evaluate their phenotypic fitness; if the fittest phenotype matches the user-defined target fitness (or other termination criterion), terminate, otherwise continue;

(iii) produce offspring by stochastic selection and recombination of the genetic material in the current population, favouring the genes of individuals with high phenotypic fitness;

(iv) introduce, with some low probability, random changes (copying errors) into the genetic material of the offspring;

(v) replace low-fitness members of the old population with the offspring created in the previous step, and return to step (ii).

This scheme is illustrated in Fig. 3.2. The selective recombination of the population's genetic material exploits information associated with good solutions to try to build even better ones, and the random mutations serve to inject entirely new and potentially favourable material, which could not have been obtained simply by recombining the genetic material of existing individuals, into the gene pool. Such an evolutionary scheme has a number of features which distinguish it from random heuristics, albeit that it might bear superficial resemblance to e.g. standard Monte Carlo approaches (Gregory, 2005).

Although early GAs encoded solutions as binary strings (as suggested in Fig. 3.2), both for the sake of simplicity and supposed theoretical optimality, today there is a large body of empirical evidence which indicates that it is almost always preferable to work directly with floating-point representations of solutions when solving numerical optimisation problems (Wright, 1991; Michalewicz, 1996; Charbonneau, 2002a; Haupt & Haupt, 2004).

It may be shown that, subject to a few reasonable assumptions, the canonical GA will always converge to the global optima in the search space in question (Eiben, Aarts & Hee, 1991; Michalewicz, 1996). This knowledge is reassuring, though usually of little practical value: of more importance is knowledge of the *rate* of convergence to the global

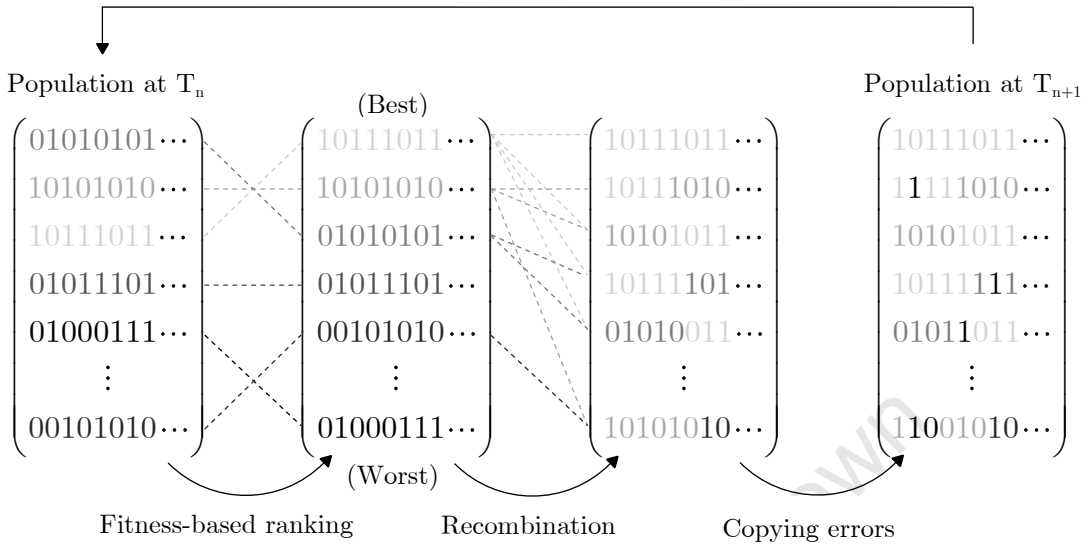

Figure 3.2 – Schematic to illustrate the workings of the canonical genetic algorithm, where trial solutions are encoded as binary strings. Each bit represents a gene; here, lighter shades are used to represent solutions determined to be fitter, according to any problem-specific metric. The 'initial' population (first array) contains a range of solutions, some poor (darker shades) and some good, or at least, less poor (lighter shades). During ranking, the solutions are sorted from best to worst (second array). The genes from fitter solutions are then favoured during recombination, which is apparent from the fact that the new solutions (third array) are assembled mainly from components of the lighter-coloured solutions. Finally, random mutations – which may or may not be beneficial – are introduced (fourth array), and the whole process repeats.

optimum. Unfortunately, with GAs, this rate tends to be highly problem-dependent; nevertheless it is possible, though usually not straightforward, to estimate the rates at which solutions are likely to be located on given problems (e.g. Holland, 1975; Thierens & Goldberg, 1994).

As already noted, most EAs do share a basic working scheme, in spite of the endless variety of their potential incarnations. Therefore, although any given EA might bear little superficial resemblance to the GA illustrated in this section (the algorithm might use a different encoding scheme, radically different mutation and recombination operators, etc.), most EAs do share the same underlying concept of a population of trial solutions evolving, under the action of a few evolutionary operators, towards optimality. ESs, for example, are similar to GAs in many respects, although usually they avoid solution encoding or discretisation, they tend to use more sophisticated mutation schemes than GAs, and crucially, they evolve the algorithm's control parameters alongside the trial solutions: an ingenious approach to meta-optimisation (Bäck et al., 1991; Kramer, 2010)!

3.4 EAs: pros and cons

This section presents some of the advantages and disadvantages associated with evolutionary algorithms. It also illustrates the performance of an EA, applied to a fairly challenging optimisation problem.

3.4.1 Some advantages

Relative to more conventional optimisation algorithms, EA-based optimisers offer a number of striking advantages, some of which are outlined below.

(i) *Robustness.* EA-based optimisers can handle – with aplomb – problems with multimodal or low-contrast objective functions, multiple objectives, and/or problems where the parameter spaces have a very high dimensionality (Charbonneau, 2002a).

(ii) *Simplicity.* The ideas underpinning EAs are intuitively accessible, and in order to solve a given optimisation problem, most 'off-the-shelf' EAs usually require little more than a single, unambiguous measure of the quality (fitness) of candidate solutions. They do not, for example, require derivatives (e.g. Jacobian or Hessian matrices), the computation of which might be prohibitively difficult or even impossible in many problems.

(iii) *Speed.* Apart from the intrinsically high speed with which EAs tend to explore large parameter spaces (Michalewicz, 1996), they are embarrassingly parallel: very little effort is required to transform a serial EA-implementation to a parallel implementation. Thus they are well-suited to exploiting high-performance hardware, e.g. multi-core workstations, graphics processing units, clusters etc. (See also the discussion in Sec. 3.6.)

(iv) *Versatility.* A single EA-based optimiser can be expected to yield 'good enough' results on a very wide class of problems – from a problem as simple as fitting a three-parameter Gaussian to some data, to one as nontrivial as choosing a molecular configuration to minimise a Buckingham potential with hundreds of parameters (Wehrens & Buydens, 1998) – and it is easy to incorporate problem-specific knowledge into an EA-based solver.

The widespread adoption of EAs in fields such as engineering, chemistry, biology, operations research, and economics bears testimony to their many merits (e.g. Charbonneau, 2002b; Haupt & Haupt, 2004), and though their uptake in the physical sciences has been somewhat slower – at least partly because the theoretical understanding of their workings is still quite limited – recently they have already been used with great success in many branches of astronomy and astrophysics (see Rajpaul, 2011, as well Sec. 3.8 of this dissertation). Indeed, if one compares the aforesaid characteristics of EAs with the many difficulties associated with microlensing modelling, as outlined in Chapter 2, it should be clear that EAs seem particularly well-suited for use in the context of microlensing modelling.

An obvious question arises: *why* do EAs work as well as they do? This topic is beyond the scope of this dissertation, but suffice it to say that a universally-accepted explanation has not yet been developed. For example, Holland's famous *Schema Theorem* (Holland, 1975) has long been touted as providing an explanation for GAs' success, although more recently it has become apparent that this theorem provides insight only into the workings of simplistic GAs, and even then, it is not clear whether the assumptions underlying the theorem are tenable (Syswerda, 1989; Wright et al., 2003).

3.4.2 Some disadvantages

Despite all their attractive features, EAs also have their share of disadvantages.

More or less in accordance with Wolpert and Macready's famous 'no free lunch' theorems (see Sec. 3.7), EAs might be called 'Jacks of all problems, but masters of none'. For example, they tend to be better at locating than at fine-tuning solutions: once an EA is in the vicinity of a global optimum, it is usually a good idea to let a local optimiser take over (Charbonneau, 2002a).

Optimising a given EA's performance to solve a given problem is often very difficult,[4] a problem compounded by the nearly endless scope for algorithmic customisation, and in order to achieve near-optimal performance it is usually preferable to hybridise an EA with problem-specific heuristics. Then again, true meta-optimisation should hardly ever be necessary with a well-designed EA, especially if one is willing to accept 'good' rather than 'excellent' performance.

EAs can be inefficient on simple problems where the computational expense of applying evolutionary operators is comparable to or outweighs the cost of evaluating the function to be optimised; conversely, on problems where each cost/fitness function evaluation is extremely expensive – for example, where each evaluation requires a long simulation to be run – an EA-based (or in fact any) forward modelling approach might be impractical.

Finally, the (currently) limited theoretical understanding of EAs is regarded, quite understandably, as a drawback by some, and this might explain their relatively slow uptake in the physical sciences.

3.4.3 Sample performance

To illustrate the performance of a typical EA, consider the following 'challenging' fitness function proposed by Charbonneau (1995):

$$f(x, y; n) = -[16x(1-x)y(1-y)\sin(n\pi x)\sin(n\pi y)]^2, \tag{3.3}$$

where $x, y \in [0, 1]$ and $n \in \{2k+1; \forall k \in \mathbb{N}\}$. For $n = 13$, say, it may be shown that $f(x, y)$ has 169 (in general n^2) local minima on its domain, only one of which corresponds to the global minimum $f(x^*, y^*) = -1$; moreover, the minima are separated by steep walls, and there is little contrast between many of the minima (see Fig. 3.3).

[4] That is to say, optimising the EA's control parameters (a process known as meta-optimisation), to suit a particular problem, is a difficult task.

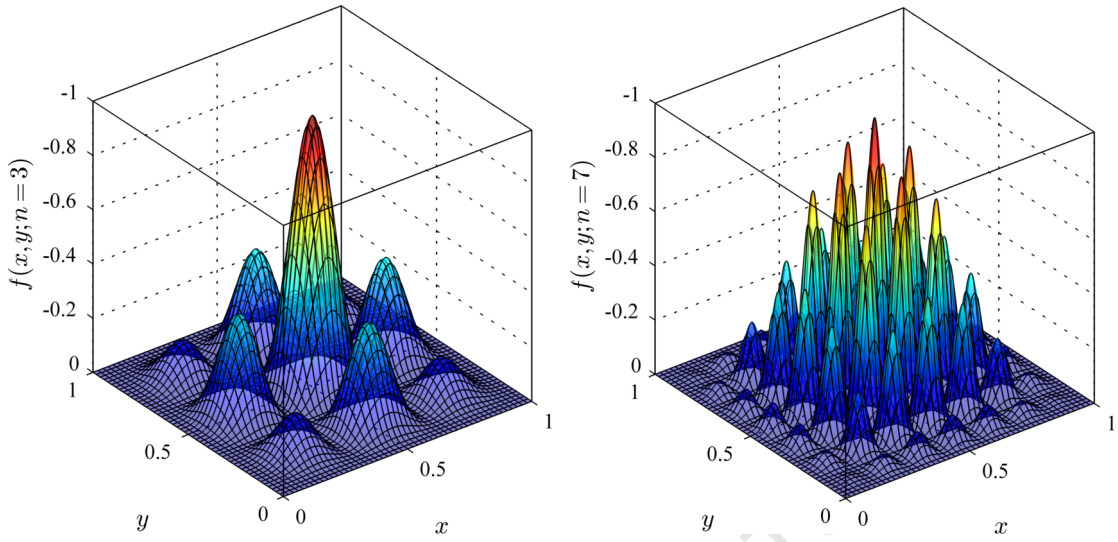

Figure 3.3 – Surface plots of the 2-dimensional function $f(x, y; n)$, defined by Eqn. 3.3, for the cases $n = 3$ and $n = 7$ (note the inverted vertical axes). In general there are n^2 local optima on the function's domain $\Omega = [0, 1] \times [0, 1]$.

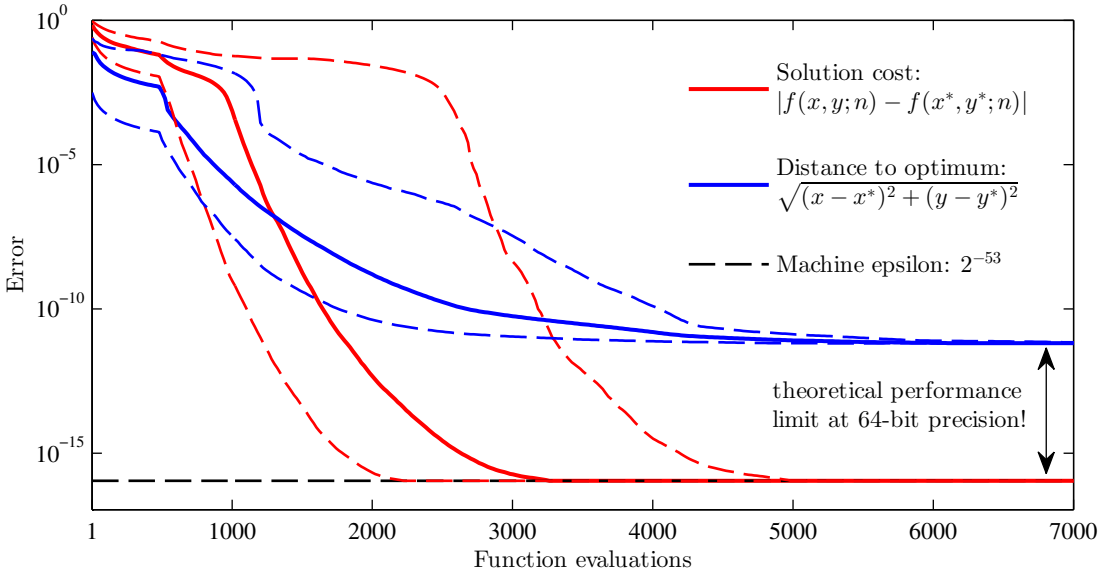

Figure 3.4 – Performance of an EA-based optimiser applied to the $n = 13$ case of the optimisation problem defined by Eqn. 3.3. The thick lines denote median performance in 10000 trials, and the thin dashed lines denote upper and lower 3σ limits (the performance differs from trial to trial on account of the stochastic nature of the EA). The global optimum is located reliably and extremely accurately after fewer than 10^4 function evaluations.

Fig. 3.4 illustrates how an EA-based optimiser fared on the rather challenging $n = 13$ problem: in 10000 trials, the algorithm converged to the global minimum every single time, with the minimum always located to an accuracy of about 10^{-10} after only $\sim 10^4$ function evaluations (or a very small fraction of a second on a modern workstation). It is worth noting that the highly nonlinear dependence of $\partial f/\partial x$ on x and y means that changes of smaller than $\sim 10^{-10}$ in x or y lead to changes smaller than $\sim \mathcal{O}(\varepsilon)$ in $f(x, y; 13)$, where $\varepsilon = 2^{-53}$ is the machine epsilon, so it is actually impossible – at least, working at 64-bit double precision – to optimise x or y to an accuracy of better than $\sim 10^{-10}$. (In fact, this is a feature common to many nontrivial optimisation problems, viz. that the objective function does not always provide useful information about the parameters to be optimised, be it because of noise, limited numerical precision, or model degeneracy.)

In any event, to put the EA's performance in perspective: a blind random search would have required $\sim 10^{20}$ evaluations to guarantee similar accuracy; on the other hand a steepest-descent (or similar gradient-based) optimiser would have had a $\sim 0.3\%$ chance of success if starting at a random point on the unit square. It is also worth emphasising that it took mere minutes to adapt an existing EA-based optimiser (viz. the EA presented in Chapter 4) to solve this problem, and that the algorithm control parameters were not optimised in any way for this new problem.

3.5 Dealing with multimodality

Though the search space defined by Eqn. 3.3 presents many challenges to optimisation algorithms (including multimodality and low contrast; see Fig. 3.3), it does at least have the convenient property that the global optimum can in principle (after an exhaustive search, say) be uniquely and unambiguously identified as the 'correct' solution to the search/optimisation problem. Unfortunately, however, with many real-world problems – microlensing modelling included – one usually cannot hope to arrive at a unique or clear-cut, globally-optimal solution, and it is instead important to sample *all* possible optima in the multimodal parameter space. In the case of modelling a microlensing lightcurve, for example, the combination of noisy observations, nonlinearity, and parameter degeneracy means that the 'correct' solution need not even correspond to the globally optimal one.

Fortunately, even if an EA is not specifically set up to search for multiple solutions, mutation operators ensure exploration of the entire parameter space (provided the evolutionary sequence is sufficiently long) – as such, given a completed evolutionary sequence, multiple possible solutions can be obtained simply by extracting all explored phenotypes that meet some fitness criterion (as opposed to extracting only the fittest phenotype in the final population state, which should correspond to a global optimum). In this way it is possible to construct and study distributions of the likely values of model parameters, rather than simply accepting at face-value the single 'best' solution found by the algorithm.

Moreover, if one knows beforehand that one is dealing with multimodal search spaces,

straightforward modifications can be made to EAs to make them more efficient at searching for multiple solutions. A simple approach is to begin a new evolutionary sequence whenever the population starts to stagnate around an optimum (see Chapter 4); alternatively, multiple independent populations can be evolved concurrently. A more sophisticated approach might be to dedicate a fraction of the evolutionary population(s) to global exploration of the search space, and the remaining fraction purely to the exploitation of promising solutions already discovered (Tsutsui et al., 1997).

3.6 On the parallelisation of EAs

As already noted, EAs are highly amenable to parallel implementations, and they parallelise especially well when the cost of evaluating candidate solutions is high (as is the case with microlensing modelling). Multi-core and multi-processor computers, clusters, grids, cloud computing platforms, and affordable general-purpose GPUs (with upwards of thousands of stream processors) are becoming increasingly prevalent in modern computing, and EAs are ideally suited to exploit such devices and platforms.

For example, by implementing an EA on a fairly high-end (though hardly top of the range) GPU, Maitre et al. (2009) managed to achieve a speed-up of roughly 100× relative to the same EA running on a standard 3.6 GHz processor, and advocated the use of multiple GPU cards to achieve even more dramatic speedups. A number of other authors have reported similar results (see Risco-Martn et al., 2012, and references contained therein): Pospichal, Jaros & Schwarz (2010), for example, managed to speed up a GA by a factor of nearly ten thousand, again using only fairly modest GPU hardware.

Furthermore, while EAs are not unique in being easy to parallelise (e.g. brute-force grid search algorithms parallelise trivially!), EAs have the advantage of being amenable to a number of *different* parallelisation schemes – some straightforward, some more sophisticated – and this provides scope for optimising an EA's performance for a specific hardware configuration and/or problem (Haupt & Haupt, 2004). The most trivial of all parallelisation schemes is to parallelise only the evaluation, each generation, of candidate solutions, with the rest of the algorithm, which will usually have a very small computational footprint relative to the evaluation of candidate solutions, left in serial form. Since selection occurs globally and within a single population, such EAs are usually referred to as 'panmictic' or 'micro-grained'. It was such an approach that was adopted for the EA developed for this dissertation (see Chapter 4); note, however, that with problems where the computational cost of evaluating candidate solutions is relatively *inexpensive*, the cost of applying evolutionary operators cannot be ignored, and efficient parallelisation would be harder to achieve.

Another popular approach for parallelising EAs is to assign to each available processor an entire evolutionary population, and then to allow these populations to evolve more-or-less independently, perhaps allowing periodic 'communication' (e.g. in the form of migration) between the populations. These so-called 'coarse-grained' or 'island' EAs are ideally suited for implementation on distributed memory MIMD (multiple instructions, multiple data) machines; in fact, even with all else being equal, island EAs tend to out-

perform single population EAs (Gordon & Whitley, 1993), with premature convergence being less of an issue and optimal solutions usually being located more quickly. In this sense, parallelising an EA can lead not only to improved computational efficiency, but also to a more effective search algorithm!

'Cellular' or 'fine-grained' EAs form the third major class of parallel EAs, and these EAs can be thought of as being intermediate to the two aforementioned classes (Lim et al., 2007). Here, the evolutionary operators are decentralised, and each processor handles a number of very small populations (perhaps containing only one or two individuals), each of which can interact with a number of 'nearby' populations. Such an implementation is a often a natural choice on an SIMD (single instruction, multiple data) computer.

3.7 On the 'no free lunch' theorems

In the field of search and optimisation, there is a set of famous theorems, due to Wolpert & Macready (1997), usually referred to as the 'no free lunch' (NFL) theorems, following the popular adage 'there ain't no such thing as a free lunch'. The NFL theorems themselves are mathematically technical, but the main practical consequence of the theorems is that the performance of any two search/optimisation algorithms will be equivalent when their performance is averaged across *all* possible problems (Wolpert & Macready, 2005). It follows that if an algorithm achieves superior performance on some problems, it must pay with inferior performance on other problems.

A common but somewhat misleading interpretation of the NFL is that a 'good' (better than average), general-purpose optimisation algorithm cannot exist, and that obtaining good performance on a given problem requires an algorithm specifically tailored for that problem. Given such a (mis)interpretation, it would be easy for a skeptic to question or dismiss the supposed advantages offered by EAs. Fortunately for EAs, however, there are many reasons why such an interpretation of the NFL Theorems is not valid in practice (Ho & Pepyne, 2002).

Firstly, the assumptions underpinning the NFL theorems do not make provision for the fact that some algorithms re-evaluate candidate solutions more often than other algorithms; regardless of specialisation to a particular problem, an algorithm that unnecessarily re-evaluates candidate solutions will be less efficient than an algorithm that does not do so. This re-evaluation might happen in an exact or even only in an approximate sense: for example, given sufficiently smooth search spaces, it will usually be inefficient to evaluate any solutions within a certain neighbourhood of solutions already known to be sub-optimal.

A more important though more subtle point, however, is encapsulated by the fact that the NFL theorems apply only when algorithmic performance is averaged over *all* possible problems, including those whose search spaces comprise only white noise, say. In fact, almost all possible problems feature objective functions or search spaces of such high Kolmogorov complexity (corresponding to extreme irregularity and unpredictability; see e.g. Li & Vitanyi, 1997) that they cannot arise in the physical world. Indeed, typical Kolmogorov-random objective functions contain more information than the observable

universe is capable of registering! For example, if, in a particular problem, candidate solutions are encoded as a sequence of 300 binary digits (bits), with the fitness values also restricted to values of 0 or 1, the associated objective function will alone contain at least 2^{300} bits of information, which is already twice the estimated bound on the number of bits that the (observable) universe can register, viz. $10^{90} \sim 2^{299}$ bits (Lloyd, 2002).

3.7.1 Real-world problems and the NFL theorems

Unlike the search spaces mentioned above, problems which have some basis in reality (a model of a physical system, say) are invariably *not* Kolmogorov random; on the contrary, their objective functions are usually highly compressible (far from random), with their topography featuring at least some degree of regularity, predictable structure, and/or continuity, all of which can in principle be exploited to expedite searches (Droste et al., 2002). For example, with such objective functions, knowing something about the value – or the slope, or the curvature, etc. – at one point usually allows us to infer some things about the functions at a family of nearby points, without actually having to visit all those points. It is certainly *not* the case that all algorithms perform equally well on such compressible functions, and indeed the set of 'realistic' optimisation problems is a vanishingly small subset of the uncountably infinite set[5] of all optimisation problems. Therefore, when dealing with real-world optimisation problems, one cannot expect the NFL theorems to apply.

For example, it is well-known that in very many real-world optimisation problems, the basins of convergence around optima can be well-approximated as quadratic bowls, and there are a number of popular gradient-based algorithms that do, in fact, exploit this property to achieve better than average (local) optimisation performance on real-world problems (Press et al., 2007). Their performance might be terrible when confronted with Kolmogorov-random functions, for example, but this is generally of little concern in practical applications. Similarly, it is entirely permissible for some *global* optimisation algorithms (including EAs) to achieve better than average performance by exploiting certain features inherent in typical, real-world objective functions. As such, the NFL theorems do *not* indicate that it is futile to try to solve an optimisation problem with an unspecialised, general-purpose algorithm.

3.7.2 Problem-specific algorithms

To be sure, incorporating prior knowledge of a problem into an algorithm, or tailoring an algorithm to a specific problem, will generally lead to an increase in the speed at which the algorithm can locate solutions to that problem. However, incorporating prior knowledge does not yield *significant* performance gains on many problems, and in any event, one often has relatively little prior knowledge with which to work.

More importantly, if an algorithm can perform optimisation in a reasonable amount of time, the manual incorporation of problem-specific information would be, if nothing else, an unnecessary expense of human time and effort (which, in the modern era, is

[5] An infinite set is said to be countable if it has the same cardinality as some subset of $\mathbb{N}$ (Lang, 1993).

usually regarded as being far more expensive than computer time). By way of analogy, knowledge of the properties of polynomials can often be used to find some of the roots of certain classes of polynomial but, especially when dealing with higher-order polynomials, it is usually far more straightforward and efficient to find the roots using a general root-finding algorithm than to bother applying analytical results (King, 1996; Press et al., 2007).

3.7.3 Free evolutionary lunches?

On a final note, Wolpert and Macready have actually proved that 'free lunches' *are* possible specifically when using so-called coevolutionary optimisation algorithms, a specialised subset of EAs. In other words, even if one disregards the practical considerations regarding solution re-evaluation, function compressibility etc., the NFL theorems still do not apply to coevolutionary optimisation algorithms, with some coevolutionary algorithms being demonstrably superior to other optimisation algorithms (Wolpert & Macready, 2005). A discussion of coevolutionary algorithms is not within the scope of this dissertation: in short, however, the coevolutionary paradigm is based on models of the cooperative evolution of two or more species, rather than just the evolution of a single species. A number of possible schemes may be used to implement this idea, for example evolving multiple subpopulations and specifying rules to govern their cooperation, or treating a species as a subcomponent of a potential solution, with complete solutions being obtained by assembling representative members of each of the species (in such a case, fitness assignment at the species level would be defined in terms of the fitness of the complete solutions in which the species members participate). For more information, refer to Potter & De Jong (1994), and references contained therein.

3.8 Applications: astronomy and astrophysics

This section presents a small sample of the numerous and diverse applications that evolutionary algorithms have found in astronomy and astrophysics. For brevity's sake, only one or two short but representative examples have been drawn from different subfields.

3.8.1 Astrophysical dynamics

Wahde and Donner developed a method for reliably determining the orbital parameters of interacting galaxies, and applied their method to both artificial and real data (Wahde & Donner, 2001). Their method is based on an EA that searches very efficiently through a large space of possible orbits; in fact, the authors argued that EAs are ideally suited for investigations of tidally interacting galaxies, where large multimodal spaces must be searched in order to constrain a large number of model parameters. Cantó *et al.* devised an interesting variant of the canonical GA, which they applied successfully to various problems, including the challenging task of finding the orbital parameters of the planets orbiting 55 Cancri, based on radial velocity measurements of the aforesaid stellar system (Cantó et al., 2009).

3.8.2 Physical and observational cosmology

Although Monte Carlo methods seem to predominate in cosmology, EAs have already found a number of applications in the field. To mention just a few: Nesseris and Shafieloo used EAs to reconstruct the expansion history of the universe in a model-independent manner and thence, in conjunction with the so-called *Om statistic*, they derived a null test on the cosmological constant model ΛCDM (Nesseris & Shafieloo, 2010); via EAs, Allanach *et al.* were able to answer some important questions related to the discrimination of SUSY-breaking models, and in particular to quantify the measurements necessary to tell different SUSY-breaking scenarios apart (Allanach et al., 2004); and Bogdanos and Nesseris used EAs to analyse Type Ia SNe data, and to extract model-independent constraints on the evolution of the dark energy equation of state (Bogdanos & Nesseris, 2009). The latter authors noted that as a non-parametric method, EAs provide a convenient, model-independent platform for cosmological data analysis which can minimise bias due to premature choice of e.g. a dark energy model.

3.8.3 Stellar spectrum fitting

Performing fits to stellar spectra is a nontrivial but important undertaking: from fitted models, one can infer a veritable plethora of stellar properties. Baier *et al.* were able to combine radiative transfer codes with an EA to produce an automated procedure for fitting the dust spectra of AGB stars. Their EA-based routine dramatically improved extant fits made with more traditional methods, and provided a quantitative platform from which to compare different models (Baier et al., 2010). In a similar vein, Mokiem *et al.* used a parallelised GA as the basis for an autonomous fitter of spectra of massive stars with stellar winds (Mokiem et al., 2005).

3.8.4 Gravitational lens modelling

Although one of the goals of this dissertation is to provide proof of concept for the use of EAs in the context of microlensing modelling, EAs are not entirely novel in the broader context of gravitational lensing. For example, Liesenborgs *et al.* presented an EA-based, non-parametric technique for inferring the projected lensing-mass distributions in strongly lensed systems (Liesenborgs et al., 2006). In fact, a genetic algorithm has already been used to model a binary-lens microlensing event (Kubas, 2005; Kubas et al., 2005); however, the algorithm used in the aforesaid work was a pre-existing learning/pedagogical tool, rather than a high-performance algorithm. (Moreover the performance of the algorithm was neither quantified nor compared to that of any other algorithms, and the study was restricted to a single event, so one cannot infer from that work that EAs are well-suited to modelling microlensing events in general.)

3.8.5 Stellar structure modelling

Metcalfe and Charbonneau implemented a highly parallelised and distributed GA to determine the globally optimal parameters for stellar structure models. The efficient,

parallel exploration of parameter space made possible by their GA-based optimisation led to some important results in the field of white dwarf astroseismology, including the unexpected resolution of a then-puzzling discrepancy between stellar evolution model and astroseismic inferences of He-layer masses in DBV white dwarfs (Metcalfe et al., 2000).

3.8.6 Telescope scheduling

Autonomous telescope scheduling is a difficult task that requires dynamic adjustment of numerous observational constraints, whilst trying to ensure the efficient achievement of many different scientific objectives. Kubanek developed a robust and easy-to-implement approach to solving robotic telescope scheduling problems, based on an EA that seeks out Pareto-optimal telescope schedules (Kubanek, 2008).

3.9 Conclusions

This chapter introduced nonlinear optimisation and evolutionary algorithms, and outlined some of their strengths and weaknesses. It was noted that EAs seem particularly well-suited to modelling microlensing events, and that the performance advantages claimed by EAs are not at all in conflict with the famous 'no free lunch' theorems. Finally, some examples were presented of the many applications that evolutionary algorithms have already found in astronomy and astrophysics.

For the reader interested in learning more about evolutionary algorithms, there are books and papers that cover nearly every imaginable aspect of the subject. To mention just a few good references, Michalewicz's book (Michalewicz, 1996) provides an excellent introduction with a theoretical leaning, while Haupt's book provides an equally good though more 'hands-on' introduction to EAs (Haupt & Haupt, 2004). Goldberg's seminal book (Goldberg, 1989), one of the most widely-cited works in all of computer science, serves as an outstanding tutorial-style reference. For more information about evolution strategies, the papers of Bäck & Hoffmeister are good starting points (e.g. Bäck & Hoffmeister, 1990; Bäck et al., 1991). Finally, Charbonneau's paper (Charbonneau, 2002a) is worth consulting: it provides a straightforward discussion of how standard statistical methods can be used to construct confidence intervals for model parameters fitted using an EA, and it also points to a number of excellent references on EAs in general.

4

An evolutionary algorithm for microlensing

This chapter is devoted to a discussion of the evolutionary algorithm developed by the author of this dissertation for the purpose of modelling microlensing events. To avoid the mouthful of 'an evolutionary algorithm designed for modelling microlensing events', the algorithm will be dubbed EMMA, an acronym for 'Evolutionary Microlensing Modelling Algorithm'.

Because the algorithm is robust enough to solve general nonlinear optimisation problems, and also because much of its development was informed by tests carried out on a broad class of optimisation problems (rather than just microlensing modelling problems), much of the discussion in this chapter is left in accordingly general terms.[1]

Sec. 4.1 gives a fairly detailed overview of the mechanics of the algorithm itself; Sec. 4.2 provides a discussion of and motivation for the features covered in the previous section; Sec. 4.3 contains some remarks on EMMA's computational implementation, and finally Sec. 4.4 discusses how EMMA can be deployed to solve a typical search/optimisation problem.

4.1 Overview of the algorithm

Note that this section simply details the mechanics of the algorithm, without providing a rationale for its various features; for the latter, refer to the next section (Sec. 4.2).

The central dynamical quantity used by EMMA, as with most EAs, is an (encoded) *population matrix*, which we denote $\mathbb{P}$:

$$\mathbb{P} = \mathbb{P}(t) = [p_{i,j}]_{N_{\text{pop}} \times N_{\text{par}}}. \tag{4.1}$$

$\mathbb{P}(t)$ is constructed so that $p_{i,j} \in [0, 1]$ *encodes* the value of the j^{th} physical parameter associated with the i^{th} individual (trial solution) – in an N_{pop}-member evolutionary population – denoted $a_{i,j}$, at generation $t \in \{1, 2, \ldots, N_{\text{gen}}\}$. By default, $N_{\text{pop}} = 1000$; N_{par} will be specified by the problem at hand.

$\mathbb{P}$ should be thought of as representing an encoded ensemble of trial solutions, with a row vector $p_{i,*}$ representing a single trial solution; an individual matrix element, $p_{i,j}$, should be thought of as representing a small 'chunk' or component of a trial solution.

[1] The algorithm's name reflects the motivation and context for the algorithm's development, but not the total scope of its possible applications.

The relationship between $\mathbb{P}(t)$ and its decoded/physical counterpart, denoted $\mathbb{A}(t)$, can be represented thus:

$$[p_{i,j}] = \mathbb{P} \underset{\text{decoding}}{\overset{\text{decoding}}{\rightleftarrows}} \mathbb{A} = [a_{i,j}] \tag{4.2}$$

The basic evolutionary scheme used by EMMA is as follows:

- Initialise $\mathbb{P}(t = 1)$.
- Iterate $t = 1, 2, \ldots, N_{\text{gen}}$; for each t:
  - (i) decode $\mathbb{P}(t) \rightarrow \mathbb{A}(t)$, and pass $\mathbb{A}(t)$ to an external fitness function;
  - (ii) rank $\mathbb{P}(t)$ based on the fitness returned by the external function;
  - (iii) select solutions for reproduction;
  - (iv) produce offspring and insert them into $\mathbb{P}(t)$;
  - (v) apply jump and creep mutations to $\mathbb{P}(t)$, and
  - (vi) check whether the evolution has stagnated (if so, make appropriate adjustments to the algorithm).
- Collate and output results, as required.

This scheme is conceptually straightforward and most of its features overlap broadly with those of the canonical genetic algorithm; many of its finer details and features – some unique to EMMA, some inspired by other EAs – are, however, non-canonical. The 'nuts and bolts' of the algorithm are described in the subsections below.

4.1.1 Initialisation

Initialising the population matrix entails simply assigning random variates with a standard uniform (rectangular) distribution, $\mathcal{U}(0, 1)$, to half the elements in $\mathbb{P}$. The remaining elements are then assigned the complements of the values already assigned. That is to say, if one element in the population matrix is assigned the value $X \in [0, 1]$, another random element will be assigned the value $(1 - X) \in [0, 1]$.

4.1.2 Encoding/decoding

The following bicontinuous linear transformation is used to relate a physical parameter value, $a_{i,j}$, to its encoded counterpart, $p_{i,j} \in [0, 1]$:

$$a_{i,j} = \alpha_j + (\beta_j - \alpha_j)p_{i,j}, \tag{4.3}$$

where it assumed that the physical parameter a_j can be restricted to some domain $\Omega_j = [\alpha_j, \beta_j]$, with α_j and β_j known. For example, if a_j is an angle, a typical domain might be $\Omega_j = [0, 2\pi)$. Even if a_j is in principle unbounded, physical considerations and/or prior knowledge associated with the problem at hand should always allow bounds

to be placed on the parameter (see also the discussion in Sec. 4.2.2). For example, if a_j represents a lensing binary-lens mass ratio, a conservative choice might be $\Omega_j = [10^{-6}, 1]$, say.

The interpretation of the encoded value $p_{i,j}$, as implicitly defined by Eqn. 4.3, is straightforward: $p_{i,j}$ represents the fractional position of $a_{i,j}$ along its 'physical' domain $[\alpha_j, \beta_j]$.

It should also be clear that EMMA works with real-valued representations of parameters ($p_{i,j} \in \mathbb{R}$) rather than discretising the parameter domains in any way, as would have been done with e.g. a binary-coded genetic algorithm.

4.1.3 Fitness evaluation and ranking

Fitness evaluation entails passing the decoded trial solutions in $\mathbb{A}(t)$ to some external, problem-specific function that assigns a well-defined figure of merit, or fitness, to each solution. In the case of data modelling, this figure of merit will usually be related to a normalised fitting error – e.g. a χ^2 statistic (or, more specifically, something like $1 - \chi^2$ or $1/\chi^2$ so that, as per the convention for EAs, better solutions can be associated with a higher fitness).

Once the solutions have been evaluated, the rows of the encoded population matrix $\mathbb{P}(t)$ are sorted so that the fittest solution occupies row 1, the next fittest solution row 2, and so on, with the worst solution in row $N_{\rm pop}$. $\mathbb{A}(t)$ is not sorted as it will not be used again during generation t; it will simply be recomputed during generation $t + 1$.

4.1.4 Selection

Once the quality of each trial solution has been ascertained, EMMA uses this information to select the solutions that will reproduce, i.e. the solutions that will have their 'genetic material' (encoded chunks of solutions) propagated to the next generation.

EMMA picks reproductive pairings via a variant of the well-known 'tournament selection' mechanism (Miller & Goldberg, 1995). The scheme used to choose any parent is as follows: *draw a random sample of* $N_{\rm tourn} \in \mathbb{N}^*$ *values from the set* $\{1, 2, \ldots, N_{\rm pop}\}$. *Find the minimum value in that sample and let this number be the* rank *of the parent chosen for reproduction.* This scheme is repeated as many times as is necessary to produce the desired number of reproductive pairings.

By default, a value of $N_{\rm tourn} = \lceil N_{\rm pop}/25 \rceil$ is used.

4.1.5 Reproduction

Offspring are produced by combining the genetic material of parent solutions. For simplicity it is assumed that exactly two parents are involved in any reproductive pairing, and that such a pairing gives rise to exactly two offspring. Assuming the parent solutions chosen for a particular pairing are $p_{i,*}$ and $p_{j,*}$, EMMA produces two offspring via the following two convex combinations of parent solutions:

$$p_{i,*} \cdot \mathbb{F} + p_{j,*} \cdot (\mathbb{I} - \mathbb{F}), \tag{4.4}$$

and

$$p_{j,*} \cdot \mathbb{F} + p_{i,*} \cdot (\mathbb{I} - \mathbb{F}), \tag{4.5}$$

where $\mathbb{F} = [f_{i,j}]$ is an $N_{\text{par}} \times N_{\text{par}}$ diagonal 'weighting' matrix, with all elements constrained to the unit interval, and $\mathbb{I}$ is the $N_{\text{par}} \times N_{\text{par}}$ identity matrix.

The actual choice of $\mathbb{F}$ determines the characteristics of the reproduction. For example, if $f_{i,i} = \frac{1}{2}\ \forall\ i$ (and $f_{i,j} = 0\ \forall\ i \neq j$, since $\mathbb{F}$ is assumed to be diagonal), the offspring will simply be arithmetic averages of the parent solutions, as may easily be confirmed via Eqns. 4.4 and 4.5; on the other hand, if each $f_{i,i}$ is assigned randomly, then the offspring will be correspondingly random interpolants of the parents solutions.

EMMA assigns $\mathbb{F}$ as follows. *For each reproductive pairing, with equal probability, assign diagonal elements of the weighting matrix* $\mathbb{F}$ *according to any one of the following three distributions*:

$$\begin{cases} f_{i,i} \sim \mathcal{U}(0,1), \text{ or} \\ f_{i,i} \sim \text{Bin}(1, \frac{1}{2}), \text{or}^2 \\ f_{i,i} = 1\ \forall\ i. \end{cases} \tag{4.6}$$

The distribution used to make the assignment is chosen randomly for each reproductive pairing. In the first case, the genes of the offspring will be random *interpolants* of the parental genes; in the second case, the offspring's genes will be randomly drawn from the combined gene-pool of the parents, but no interpolation will occur; and the final case corresponds to asexual reproduction (Cantó et al., 2009) wherein offspring are identical copies of their parents (useful for giving the upcoming mutation operators multiple opportunities to fine-tune a solution that is already very good). Both the second and third schemes can be thought of as limiting cases of the first, more general scheme.

All solutions in $\mathbb{P}(t)$, save for the fittest one, are replaced by the newly-created offspring.

4.1.6 Creep and jump mutation

All newly-created offspring are subject to fine-grained mutations, also known as 'creep mutations'. These mutations randomly increase or decrease an encoded value by a randomly-chosen step with a log-uniform distribution:

$$p_{i,j} \rightarrow p^+_{i,j} \text{ or } p^-_{i,j}, \quad \left\{ \begin{array}{l} p^-_{i,j} = p_{i,j} - (p_{i,j} - 0) \cdot X, \\ p^+_{i,j} = p_{i,j} + (1 - p_{i,j}) \cdot X \end{array} \right\} \tag{4.7}$$

where the actual assignment ($p^+_{i,j}$ or $p^-_{i,j}$) is chosen with equal probability, $\ln X \sim \mathcal{U}(\ln \varepsilon, \ln 1)$, and ε is the machine epsilon. Note that the creep mutations are fully compatible with EMMA's parameter encoding scheme since $\forall\ X, p^{\pm}_{i,j} \in [0,1]$.

[2] Here, Bin(n,p) is the discrete binomial distribution for n trials, with a probability p of success. Note that for $n = 1$, the distribution reduces to the Bernoulli distribution, which assumes the value 1 with probability p, and 0 with probability $1 - p$ (Bevington & Robinson, 2003).

In addition to the creep mutations, the offspring are exposed to the possibility of coarse-grained or 'jump' mutations. With a jump mutation, an element of the population matrix is 'flipped' to a completely random value on the unit interval:

$$p_{i,j} \to X \sim \mathcal{U}(0,1). \tag{4.8}$$

A jump mutation is assumed to happen with some small probability $0 < p_{\text{mut}} \ll 1$, so that in a given generation, on average $p_{\text{mut}} \times N_{\text{pop}} \times N_{\text{par}}$ elements of $\mathbb{P}(t)$ are subject to a jump mutation. By default, $p_{\text{mut}} = 0.01$ is used, meaning that each generation, 1% of the elements $\mathbb{P}$ will be randomly changed.

4.1.7 Stagnation checks

If the evolution is deemed to have stagnated – as determined by any criterion, the default one being the fitness of the fittest individual in $\mathbb{P}(t)$ not having improved by more than 1% over the past 10 generations – the jump-mutation probability will be boosted (increased by 50%); conversely, if the evolution has *not* stagnated, the jump-mutation probability will be throttled back (decreased by 50%).

As a further step, if repeated boosts to the jump-mutation probability do not have the desired effect, i.e. stagnation flags are raised repeatedly, $\mathbb{P}(t)$ will be saved to memory, and the population will be reinitialised as was done at $t = 1$, so that $\mathbb{P}(t+1)$ will be a fresh evolutionary population with no dependence on $\mathbb{P}(t)$.

4.2 Discussion of EMMA's features

This section provides a discussion of, and motivation for, the various features outlined in the previous section.

4.2.1 Initialisation

The initialisation scheme used by EMMA is a simple and very popular scheme used to initialise EAs; it serves to ensure a uniform, unbiased, and diverse initial sampling of the entire (encoded) parameter space (Haupt & Haupt, 2004).

Of course, if one has some prior information about the probable distribution of parameters in the problem being solved (e.g. first-order guesses, or probability distributions over the parameter ranges), intuition suggests that this information could be incorporated into the evolution via a bias in the initial population – that is, a non-uniform initial sampling of parameter space, perhaps with many trial solutions clustered around a first-order guess at the solution. In practice, however, it is preferable to incorporate such information via two possible alternative schemes, *instead* of biasing the initial population:

(i) by modifying the default parameter encoding scheme, which assumes that all possible values on the parameter domains are initially equally probable (for example, the parameter encoding could easily be adjusted so that more probable parameter

values are assigned proportionately larger regions of encoded parameter space, so that certain subsets of parameter space are excluded entirely, etc.); and/or

(ii) by modifying the fitness function (see the discussion in Sec. 4.2.3 about Bayesian likelihood functions).

A primary motivation for *not* biasing the initial population is that whereas a few dramatic mutations, or the early predominance of suboptimal solution, say, could quickly lead to the destruction of whatever useful prior information was contained in the distribution of the initial population, with either of the above approaches, the prior information may be utilised for all t.

4.2.2 Encoding/decoding

The decision to use a real-valued representation of solutions, given the algorithm's primary goal of fitting microlensing lightcurves, was an easy one to make. Binary encoding *was* favoured with early genetic algorithms, and there are of course many discrete optimisation problems – including classic problems such as the travelling salesman problem and the boolean satisfiability problem, both of which were used extensively for benchmarking early genetic algorithms – where binary encodings provide a very natural fit (Michalewicz, 1996). However, there is a large body of empirical evidence which indicates that real-coded or 'continuous' EAs lead to superior performance on real-valued numerical optimisation problems (Wright, 1991; Houck et al., 1996; Charbonneau, 2002a).

Performs gains notwithstanding, real-coded EAs are also far easier to work with: it is very difficult to design and to control the characteristics of binary evolutionary operators. See e.g. Srinivas & Patnaik (1994) or Haupt & Haupt (2004) for a discussion of the myriad difficulties – including Hamming walls, unpredictable crossover operators, bloated code, and more – associated with binary-coded EAs.

EMMA's encoding scheme amounts to working with all physical parameters normalised to the *unit* interval; the normalisation, in particular to the unit interval (as opposed to any other real interval), is largely a matter of convenience. The normalisation itself means that all parameters can be treated on an 'equal footing' when applying evolutionary operators: for example, without normalisation, a different mutation operator would be required for each different parameter, to ensure mutations don't produce parameter values outside the allowable domains – whereas after normalisation, a single mutation operator can be applied to $\mathbb{P}$ as whole (making for simpler code; see discussion in Sec. 4.3). As for the normalisation specifically to the *unit* interval: the unit interval corresponds identically to the support of the standard uniform distribution (which serves as a basis for sampling from most other probability distributions), a fact that greatly simplifies the coding of the stochastic components of the algorithm.

EMMA's encoding scheme, encapsulated by Eqn. 4.3, does assume that all parameters can be restricted to simple and finite physical intervals. Certainly in the case of microlensing modelling, this assumption is not restrictive: all parameters of interest are either constrained by definition, e.g. $0 \leq \theta < 2\pi$ and $0 < q < 1$, or they can at least be restricted to sensible ranges via physical considerations, e.g. $0 < a, b \lesssim \mathcal{O}(10)$.

Even (hypothetically) unconstrained parameters can be 'tamed' via simple mathematical transformations: e.g. if $a_{i,j} \in [0, \infty)$, then $\arctan a_{i,j} \in [0, 2\pi]$. More generally, it is possible to devise useful encoding schemes that can handle parameters that are constrained to far more complex sets. Indeed, it is possible simply to forgo an encoding scheme altogether, though this would come at the cost of more complicated code for handling mutation operators, crossover operators etc.

4.2.3 Fitness evaluation and ranking

This is the most straightforward yet most important part of the algorithm. Because EMMA's selection scheme is rank-based rather than fitness-based (see next subsection), the exact choice and the normalisation of the fitness function (e.g. $1 - \chi^2$ vs. $1/\chi^2$ vs. $1 - \sqrt{\chi^2}$) is largely irrelevant to EMMA, and this greatly enhances the algorithm's robustness, as its performance does not hinge on a 'clever' choice of a fitness function. More precisely, the algorithm's performance will be identical when using any two fitness functions that can be related via a monotone (order-preserving) transformation. The only *proviso* is that the fitness function used should be able to map any solution to a real number that provides some well-defined measure, whether relative or absolute, of its quality.

An important point is that fitness-ranking can actually be performed according to arbitrary Bayesian priors and likelihood functions. In fact, the algorithm itself makes absolutely no assumptions about the statistical paradigm (e.g. Bayesian vs. frequentist) in which data modelling is taking place. Therefore EMMA may be used to perform everything from ordinary least-squares fitting (if the fitness function is chi-squared based, for example), to maximum *a posteriori* parameter estimation (using an appropriate Bayesian posterior as fitness function), to robust model parameter estimation (e.g. using L-, M- or R-estimates). A detailed discussion of the pros and cons of these different possible approaches to data modelling is given by Press et al. (2007).

It is also interesting to note that fitness evaluation is the only step during which EMMA makes any real contact with the 'external world', which is to say the physical problem being solved; all other parts of the algorithm are entirely self-contained.

4.2.4 Selection

The *tournament-style selection* used by EMMA is one of the two most popular and well-studied evolutionary selection schemes, the other one being 'roulette-wheel selection' (also known as 'proportional selection').

The basic idea behind tournament-style selection is that it corresponds to running a number of small tournaments or 'fights' between solutions. A typical tournament might comprise, say, $N_{\text{tourn}} = 4$ randomly-chosen members if $N_{\text{pop}} = 100$, with the 'victor' of an individual tournament, viz. the fittest (best-ranked) solution in that particular tournament, being allowed to reproduce. The larger the tournament, the higher the *selection pressure*, i.e. the less likely that poor solutions will be chosen for reproduction. Fig. 4.1 illustrates how N_{tourn} can be used to adjust selection pressure. A strong selection

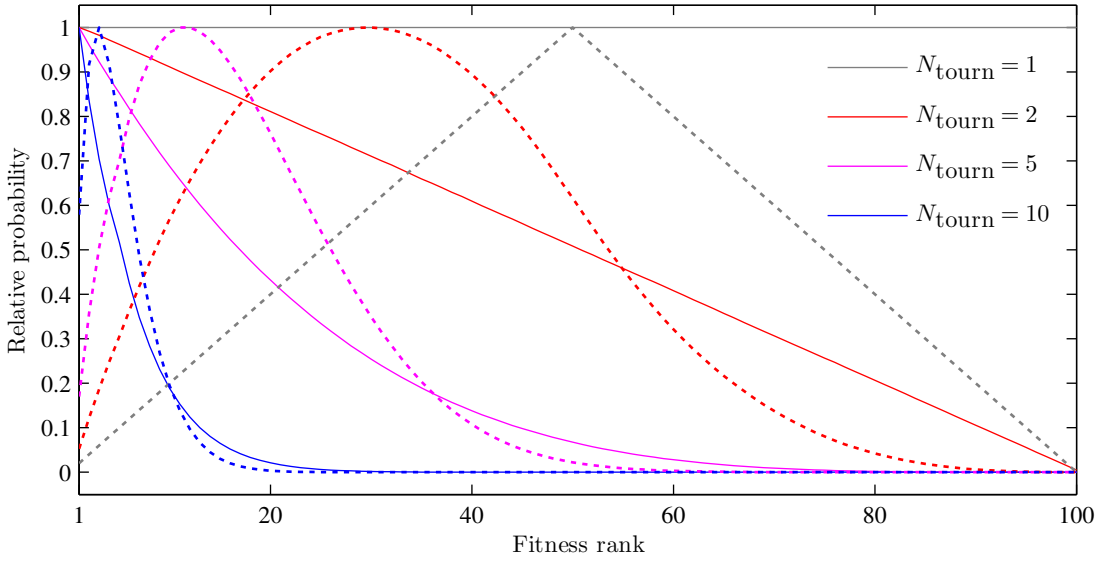

Figure 4.1 – An illustration of how N_{tourn} (tournament size) affects selection pressure in tournament selection. In this example, the population size is $N_{\text{pop}} = 100$. The solid lines show the probability distributions for the fitness of individuals selected for reproduction, and the dotted lines show the relative probability distributions for the *mean* fitness within the resultant pairings.

pressure tends to facilitate quicker local optimisation (via exploitation of existing good solutions), whilst a weak selection pressure tends to speed up global exploration (since, on average, less time will be spent exploiting good though sub-optimal solutions).

With roulette wheel selection, the probability of any solution being chosen for reproduction is proportional, linearly or otherwise, to the ratio of its fitness to that of the mean fitness of the population as a whole. It is well-known for being the selection mechanism used in the canonical genetic algorithm, although its fame probably outweighs its usefulness.

One of the main advantages of tournament-style selection is that it doesn't depend at all on the choice of a fitness function: for example, using roulette-wheel selection and a high-contrast fitness function, a relatively fit solution might prematurely come to dominate the entire population, even if that solution is actually sub-optimal; converse problems exist with low-contrast fitness functions. Another advantage is computational efficiency: tournament selection requires little more than the generation of suitably-sized sets of random numbers, and choosing the minimal values in these sets. For a detailed discussion of the relative merits of various selection mechanisms, refer to e.g. Goldberg & Deb (1991).

4.2.5 Reproduction

The scheme used to assign $\mathbb{F}$ (Eqn. 4.6) was chosen based both on empirical testing and a large body of literature that discusses the relative merits of different reproductive schemes. EMMA's reproductive scheme was designed to 'cover all bases', but it is worth adding that the performance of real-coded EAs is generally not very sensitive to the choice of reproductive scheme. For example, if EMMA's reproductive scheme is changed to something simpler but still sensible (e.g. $f_{i,i} \sim \mathcal{U}(0,1)$ only), its performance does not change dramatically. Such behaviour would certainly *not* be possible with binary-coded GAs, where an algorithm's performance usually depends very strongly on the exact choice of crossover operator. Moreover, whereas crossover (reproduction) in binary-coded GAs is actually used both for exploitation of existing solutions *and* for global exploration, EMMA's reproduction operator is more conservative, and its role is clearly restricted to exploitation.

The non-replacement of the fittest individual in $\mathbb{P}(t)$ is referred to as 'elitism' in the literature (e.g. Michalewicz & Fogel, 2000), and is a standard feature of most EAs. A practical implication of elitism is that the fitness of the fittest individual in the population will increase monotonically with t (unless of course the evolutionary sequence is restarted, as described in Sec. 4.2.7: even in this case, however, the evolution history of the population is saved, so in a global sense, the fitness of the best solution known, as a function of t, will still increase monotonically).

4.2.6 Creep and jump mutation

The most well-known problem associated with the canonical GA's 'bit-flip' mutation operator is that very often, the flipping of just a single encoded bit will lead to a drastic jump in all parameters in the decoded solution. This is might be a desirable characteristic early on in an evolutionary sequence, when one wants to explore parameter spaces in a global sense; however, when one wishes to improve upon existing good solutions, these so-called jump mutations will be extremely destructive because, more often than not, they will lead to inferior solutions and/or solutions outside the basin of convergence of the existing solution(s). This problem will only be exacerbated as one move closer to the true solution. Thus, jump mutations are antithetical to local optimisation – and yet they are a necessary evil, not only for early global exploration, but also as a continual safeguard against a population getting stuck in the basins of attraction of *suboptimal* solutions.

A very large body of literature (see Hong et al., 2002, and references contained therein) has been devoted to the design and study of improved mutation operators, and most of these non-canonical mutation operators are based loosely on the idea of replacing jump mutations with 'creep' mutations, i.e. controllably small (in some well-defined sense) changes to existing solutions, instead of very large ones. For example, one simple and intuitively-appealing scheme, inspired by physical annealing processes, is to allow larger (jump-like) mutations early in the evolution, but to have the maximum allowed mutation size 'decay' with t, so that the role of the mutation operator segues from global

exploration to local optimisation. Such an approach can, however, lead to populations becoming permanently stuck in the basins of suboptimal solutions.

The mutation operators outlined in Sec. 4.1.6 were designed to strike a balance between adequate global exploration and efficient local optimisation, and their design was informed both by extensive empirical testing and by theoretical considerations. For example, the creep operator was designed to favour relatively small perturbations (the log-uniform distribution is a standard assumption for any variable that can span many orders of magnitude, with little else being known about its underlying distribution). Mutation sizes were deliberately bounded below by $\mathcal{O}(\varepsilon)$, and above by $\mathcal{O}(1)$, respectively in order to prevent wasteful mutations (any mutation smaller than ε would be rendered meaningless by floating-point roundoff), and to ensure full compatibility with the parameter encoding scheme. The self-adaptation of mutation *probabilities* is explained in the next subsection.

4.2.7 Stagnation checks

The motivation for the autonomous boosting/throttling back of mutation probabilities, in response to the rate of improvement of solution fitness, is straightforward. If the algorithm is succeeding at making significant (at least, in a relative sense) and continual improvements to the fitness of existing solutions, it is safe to assume that one or more members of the population are already in the basin of attraction of an optimum (one or two sequential improvements may be ascribed to serendipity, though not so with improvements correlated over a number of generations). In such situations, it is desirable to suppress jump mutations, and to favour local optimisation via the more conservative recombination and creep mutation operators. On the other hand, if steady improvements are no longer being made, either local optimisation has concluded (i.e. the bottoms of the wells of the aforementioned optima have been located), or there are no members of the population in the basin of convergence of an optimum. Whichever the case, emphasis will need to shift back to global exploration, hence the boosting of the jump-mutation probabilities.

The inclusion of a mechanism to restart the evolution sequence when it stagnates yields similar benefits to evolving many independent populations concurrently over a relatively short number of generations. The latter idea has already been tested extensively in the literature (Haupt & Haupt, 2004), and as such, EMMA's 'restart mechanism' represents a straightforward modification to a well-established idea. In any event, two key advantages offered by EMMA's restart mechanism (or indeed by a multiple-population approach) are that:

(i) the algorithm will not get stuck exploiting good but suboptimal solutions; and

(ii) if there are no obviously 'correct' or globally optimal solutions, the algorithm will at least be more likely to locate all potential solutions (whether locally or globally optimal).

The restart mechanism is also arguably conceptually simpler – and certainly easier to implement computationally – than the multiple-population approach.

Of course the restart mechanism means that the idea of steady converge to a single optimum needs to be discarded. Even though the fitness of the best known solution will increase monotonically over time (by virtue of both elitism and the fact that the population is saved to memory before a restart), EMMA should be thought of not so much as an algorithm that will provide a single 'final answer', but rather an intelligent sampling of the entire parameter space; contained within that sampling will hopefully be one manifestly superior (globally optimal) solution, or at least multiple potential solutions. This is very similar to the output one would expect to obtain at the end of an MCMC run (Gregory, 2005) – the difference being that with EMMA, good solutions will be indicated not by the most densely-sampled regions of parameter space, but rather by their fitness values.

4.2.8 General design considerations

As suggested in Sec. 3.4.2, EAs tend to have virtually infinite scope for customisation, so there is little hope of ever truly optimising the design of any EA (even for some fairly limited class of problems). The aim with EMMA was simply to design a robust, efficient, and versatile evolutionary algorithm that yields 'good enough' performance on a useful spectrum of problems, including modelling microlensing events. Accordingly, EMMA's general development (algorithm structure and operator design, internal parameter choices etc.) was informed by tests carried out not only on single- and binary-lens microlensing modelling problems, but also on a wide range of challenging nonlinear optimisation problems (Moré et al., 1981; Molga & Smutnicki, 2005).

Of course, meta-optimisation of EMMA's control parameters could lead to performance gains on some problems (presumably with a decrease in performance on others), but of course one has to question whether the value of spending inordinate lengths of time obtaining incremental performance gains on hypothetical problems, when one could instead be spending that time deploying the algorithm and solving actual problems.

Incidentally, it should be mentioned that EMMA did have a number of 'sibling algorithms' during the course of its development: for example, one of the more interesting variants incorporated ideas from the field of evolution strategies. In this variant, an individual's genome contained not only encoded physical parameters, but also control parameters (e.g. parameters governing mutation directions, mutation step sizes etc.) related to the evolution of the physical parameters themselves, and a *further* layer of parameters that controlled the evolution of the 'lower-level' control parameters. Even though this algorithm had the advantage of performing meta-optimisation (and meta-meta-optimisation!) seamlessly and concurrently with the primary parameter optimisation, testing suggested that EMMA managed to achieve a better balance between robustness, speed and simplicity. Other tested variants included a 'memory-based' EA where the mutations vectors in one generation were stored, and used to inform mutations in the next generation (a successful mutation direction, say, found in generation t would be favoured in generation $t + 1$); an EA whose dynamics were governed by the diversity of the ensemble of solutions in $\mathbb{P}(t)$; and an EA where offspring were produced by recombining the genes of multiple (more than two) parent solutions.

4.3 Notes on EMMA's computational implementation

EMMA was coded in MATLAB. The MATLAB language/computing environment was chosen for the following reasons:

(i) the author of this dissertation has extensive experience with the MATLAB language;

(ii) it is ideally-suited to (and highly optimised for) numerical, especially array-based, computation, unlike e.g. Python, which is a more general purpose language;

(iii) it interfaces easily with code written in other languages; and finally,

(iv) it is a 4GL (fourth generation language), so code prototyping and deployment can be carried out far more rapidly than with a 3GL, and yet its performance characteristics are akin to those of 3GLs (C, C++, Fortran etc.).

An impressive illustration of the final point is provided by Kepler's data processing pipeline (Jenkins et al., 2010): in the interests of rapid development, the pipeline was initially coded largely in MATLAB, the plan being that it would then be translated into a lower-level language. However, the performance was so impressive that it was decided that it'd not be worth translating the pipeline into another language, and the MATLAB 'prototype' became the final pipeline!

Even though EMMA was coded in MATLAB, the external fitness function – the function on which the entire algorithm will pivot – can be written in C, C++, MATLAB or Fortran, without any modification to EMMA's code. For example, in the microlensing modelling experiments described in this dissertation (Chapters 5 and 6), the external fitness functions used by EMMA were coded in C++.

EMMA was coded in such a way as to achieve maximal computational efficiency, rather than maximal clarity or user-friendliness. The rationale for this was twofold: there already exist a handful of fairly popular EA-based optimisers (e.g. PIKAIA: Charbonneau, 2002a) that are geared heavily towards user-friendliness instead of performance, and it was decided that EMMA's performance should not be compromised on difficult microlensing problems; and secondly, EMMA could very easily be rewritten so as to improve user-friendliness.

The original MATLAB implementation of the algorithm, as prepared for this dissertation, is available directly from the author. The code itself is not included in this dissertation because although the code is quite compact, it would make little sense without the inclusion of extensive documentation to explain the workings of the various external functions called by the code, MATLAB's built-in functions and idiosyncratic syntax and calling conventions, etc.; moreover, the detailed description of the algorithm given in Sec. 4.1 is sufficient to construct – in any language – a working version of the algorithm.

4.3.1 The array programming paradigm

EMMA was coded in such a way as to conform to the 'array programming paradigm' (Iverson, 1980): all of its operators are applied to a single population matrix rather than to individual solutions. This makes for more concise and efficient code.

As an example, by pre-tiling the α_j's and the β_j's (see Eqn. 4.3), respectively, into two $N_{\rm pop} \times N_{\rm par}$ matrices, and using the Schur/Hadamard product instead of the standard matrix product, EMMA is able to carry out all solution decoding with just a single line of matrix arithmetic. Similarly, by careful exploitation of array and matrix arithmetic operators, all reproduction and mutation within the population is accomplished with just a few lines of code, sans the need for any iteration statements (e.g. `for` loops).

As such, despite the apparent complexity of the EMMA, its core code comprises fewer than 100 lines, and only about half of these lines are executed repeatedly (fitness evaluation, application of evolutionary operators), with the remainder being used to declare variables, define control parameters etc. This line count does, however, exclude problem-specific, external functions called by EMMA (e.g. in the case of microlensing modelling, a function to convert trial solution parameters into lightcurves, following the formalism in Sec. 2.3).

4.3.2 Parallelisation and computational footprint

To maximise performance by fully exploiting available processing hardware, EMMA was coded so that in a given generation, each of the n available processor threads are used to evaluate the fitness of $1/n^{\rm th}$ of the total $N_{\rm pop}$ trial solutions. This high level of parallelism means that using a 6-core, 12-threaded CPU, for example, EMMA will usually be $\gtrsim 500\%$ faster, in terms of solution evaluations per second, than an identical, non-parallelised algorithm. (Although EMMA is CPU-intensive, it places very modest demands on the primary memory of a computer; evolving a typical evolutionary population usually requires no more than a few megabytes of primary memory.)

To give an idea of EMMA's computational footprint: on a nontrivial problem such as fitting simple binary microlensing lightcurves (Chapter 5), the evaluation of a population of trial solutions will usually require *at least* least two orders of magnitude more computational time than the time required to run all the code related to evolving the population of solutions. In other words, EMMA's computational footprint will be wholly negligible, and on such problems, any bottleneck in the algorithm's performance will be due to the external fitness function.

On the other hand, on relatively trivial problems – fitting a Gaussian to a spectral line, say – EMMA's computational footprint will probably be comparable to, if not greater than, that of the fitness function in question. However, using a sophisticated evolutionary algorithm to solve a simple optimisation problem would be akin to using a chainsaw to prune a small hedge: certainly, it'd get the job done, but it'd probably be easier just to use a pair of small shears!

No in-depth study was made of the performance gains that could be expected when implementing EMMA on a graphics processing unit (GPU). Given that the algorithm's computational footprint will generally be much smaller than the fitness function associated with a difficult optimisation problem, the most substantial gains could be expected simply from implementing the fitness function evaluation on a GPU, rather than porting the entire algorithm to a GPU (Maitre et al., 2009). In fact, preliminary work carried out by the author, using MATLAB's Parallel Computing Toolbox (Sharma & Martin,

2009), suggested that even with a very-low end, mobile (i.e. laptop-based) GPU, evaluation of binary microlensing lightcurves can be carried out at about the same speed as when using an ordinary central processing unit (CPU). Of course, this does not sound particularly impressive – especially given the effort required to rewrite code to run on a GPU – but it does suggest the potential for far more significant speedups, both by using higher-end GPU hardware (e.g. nVidia Tesla boards), and moreover by writing code for a dedicated parallel computing architectures, such as nVidia's Compute Unified Device Architecture (CUDA).[3] See also the discussion in Sec. 3.6.

4.4 Using EMMA to solve a problem

The minimal ingredients required for using EMMA to solve an optimisation problem are the following:

(i) a well-defined, problem-specific fitness function that maps physical parameters to a fitness/figure of merit; and

(ii) an encoding scheme (e.g. Eqn. 4.3) that maps physical parameters to the unit interval.

The algorithm can be run either until some user-defined convergence criterion is met, or until a certain length of time has elapsed, and the algorithm's control parameters can be left at their default values. Adopting this approach, it takes mere minutes to set up EMMA to solve a new optimisation problem, and even using the default parameter values, fairly good performance could be expected 'straight out of the box' (refer the example in Sec. 3.4.3).

If one wants more control over the behaviour of the algorithm, all of its control parameters (mutation rates, population sizes, selection pressure, stagnation criteria) may easily be adjusted. Beyond the basic control parameters, the algorithm as a whole can also be modified in virtually endless ways (as a simple example, it can be coupled to a dedicated local optimiser, as is done in the next chapter).

The outputs from the algorithm can also be adjusted, with minimal effort, to suit one's needs: on simple problems one might only be interested in a single 'best solution' (and the quality of that solution), whereas with more nontrivial, multimodal problems, one might wish to obtain $\mathbb{A}(t)$ for all t, and the associated fitness vectors, e.g. $\chi^2(\mathbb{A})$. The latter information could be used, say, to construct confidence intervals for model parameters (Charbonneau, 2002a).

An example of EMMA's performance on a nonlinear optimisation problem has already been presented in Sec. 3.4.3.

[3] Unfortunately, the author of this dissertation has no experience with platforms such as CUDA. MATLAB's Parallel Computing Toolbox does facilitate GPU computation by using wrapper functions to access CUDA, although at present, its performance leaves much to be desired

4.5 Conclusions

This chapter introduced and discussed the evolutionary algorithm (dubbed EMMA, an acronym for 'Evolutionary Microlensing Modelling Algorithm') that was developed by the author of this dissertation for the purpose of modelling microlensing events, as well as for solving more general numerical optimisation problems.

A fairly detailed overview was given of the mechanics of the algorithm itself, and all of the algorithm's features were motivated and discussed. A commentary was also given on the computational implementation of the algorithm, as well as on how the algorithm may be used to solve a typical problem. It was noted that the algorithm is versatile, robust, conceptually straightforward, highly customisable, and well-suited to exploiting parallel computing architectures.

5
EA-based lightcurve fitting

This chapter presents fitting experiments that were set up and run in order to provide proof of concept for the use an evolutionary algorithm to model microlensing events, and also to demonstrate the performance of EMMA, the evolutionary algorithm introduced in the previous chapter of this dissertation.

The chapter is structured as follows. Sec. 5.1 discusses the different algorithms compared in the fitting experiments. Sec. 5.2 describes the fitting experiments in question, and Secs. 5.3–5.5 present the results of these experiments. Sec. 5.6 contains an in-depth discussion of the results, and Sec. 5.7 concludes.

5.1 Algorithms compared in tests

Four algorithms are compared in the experiments that follow: a grid search algorithm, an iterated simplex algorithm, an evolutionary algorithm (EMMA), and an artificial neural network. Each algorithm is discussed briefly in the subsections below.

5.1.1 Grid search algorithm

A simple brute-force approach, this algorithm systematically evaluates a 'grid' of points, i.e. a predefined, uniformly-sampled subset of the model parameter space. The grid can be made arbitrarily fine (so that optimal solutions can, in principle, be located with arbitrary accuracy), but computation time, which increases linearly with the total number of gridpoints, places a practical limit on the number of gridpoints that can be tested.

In the microlensing fitting experiments, each dimension of the parameter space was discretised into the same number of linearly-spaced gridpoints, with the total number of gridpoints being constrained only by a predetermined limit on the fitting time. For example, if 10^6 trial solutions could be evaluated within the time limit, and there were 7 parameters to be fit, $(\ln 10^6 / \ln 7) \sim 7$ different values of each parameter would be tested.

For reasons outlined in Sec. 2.3.4, such a trivial algorithm should not be expected to be suitable for the efficient modelling of microlensing events; still, it is included in the tests that follow as a first-order foil for the more sophisticated algorithms. Moreover, and somewhat surprisingly, grid search algorithms *have* found some (admittedly limited) use

in the context of microlensing modelling (e.g. Kubas, 2005; Bennett, 2010; Gaudi, 2010), which underscores the paucity of search/optimisation algorithms, other than brute-force approaches, capable of reliably making headway on microlensing modelling problems.

5.1.2 Iterated simplex algorithm

The 'downhill simplex method', also known as the 'amoeba algorithm' or the 'Nelder-Mead method', is an enormously popular nonlinear optimisation technique, due to Nelder & Mead (1965). The downhill simplex method is used primarily for local optimisation, although it is endowed with some global optimisation capabilities, and like many other heuristic algorithms – including evolutionary algorithms – its operation is based purely on the *evaluation* of a fitness/objective function; it does not, for example, depend on numerical or analytical gradients of such a function.

Given an optimisation function in n free variables, the method starts by computing the values of the function at $n + 1$ different points that form the vertices of a general simplex.[1] Then, by extrapolating the behaviour of the function from the measurements made at each of these test points, the algorithm rapidly adapts the simplex to the local landscape, in such a way that the simplex is made to 'explore' its way downhill, until finally contracting onto a local minimum.

In the microlensing fitting experiments, the modern downhill simplex algorithm of Lagarias et al. (1998) was used.

As demonstrated by Charbonneau (2002a), the use of multiple, short simplex runs (hereafter an 'iterated simplex' approach), rather than one very long simplex run, greatly enhances performance on difficult *global* optimisation problems: accordingly, in the fitting experiments, a number of random starting points were chosen in the search space, and at each point, a new downhill simplex optimiser was deployed. An individual simplex run was terminated when convergence to a local optimum was achieved, or when 10^4 iterations had been reached. As with the grid-search approach, the number of starting points considered was limited only by the available computational time.

Given the popularity, speed and *modus operandi* of the amoeba method, an iterated simplex approach constitutes a very reasonable competitor for the evolutionary algorithm.

5.1.3 Evolutionary algorithm

This is the newly-developed algorithm (EMMA) presented in the previous chapter. In the fitting experiments, default algorithm parameters were used, viz. a fixed population size of $N_{\mathrm{pop}} = 1000$, a fixed selection-pressure parameter of $N_{\mathrm{tourn}} = 40$, and a starting jump-mutation probability of $p_{\mathrm{mut}} = 1\%$.

Although EMMA was designed to have both global exploration and local optimisation capabilities, its fits can very easily be fine-tuned by a *dedicated* local optimiser such as

[1] A simplex is a geometrical figure consisting, in n dimensions, of $n + 1$ points/vertices and all their interconnecting line segments, polygonal faces, etc. In two dimensions, a simplex is a triangle; in three dimensions, it is a tetrahedron.

the amoeba method described above. The performance of EMMA both with and without the benefit of amoeba fine-tuning was studied; in the former case, the amoeba method was used to fine-tune only one (viz. the fittest) solution returned by EMMA.

5.1.4 Artificial neural network

'Artificial Neural Networks' (ANNs, sometimes referred to simply as 'neural networks') are mathematical/computational models that are inspired by biological neural networks. In an ANN, simple computational processing elements ('artificial neurons') are joined together to form a complex network of processing nodes, mimicking the structural and functional aspects of biological neural networks. ANNs are popular in the fields of regression, classification and pattern recognition, and decision-making; possibly the most attractive feature of ANNs, however, is their ability to be used as arbitrary function approximation mechanisms that can 'learn' from observed data (Michalewicz & Fogel, 2000). In other words, they can perform autonomous, black-box modelling of the unknown (and possibly very complicated) functional relationship between a set of input and output vectors, e.g. the relationship between a set of microlensing lightcurves, $\mathcal{X}$, and their underlying model parameters, $f(\mathcal{X})$.

Given a finite but possibly very large set of so-called 'training data', $\{\mathcal{X}, f(\mathcal{X})\}$, an ANN will learn, by adapting its network structure, that when presented with input vectors $\mathcal{X}$, it should produce outputs as close as possible to $f(\mathcal{X})$; moreover, when presented with some *new* input vectors $\mathcal{X}'$ that did not form part of the original training set, the ANN will be set up in such a way as to be able to produce output vectors based on e.g. interpolation or extrapolation from its knowledge of the *original* training data. If all goes according to plan, the new outputs should be good approximations to $f(\mathcal{X}')$.

Training an ANN usually amounts to solving a (difficult) nonlinear, global optimisation problem, the goal of which is to minimise the difference between the ANN's outputs and the 'correct' outputs defined by the training set – in fact, the training of neural networks is very often handled by evolutionary algorithms. Once suitably-trained, an ANN will usually be able to map input vectors to output vectors very rapidly. Unfortunately, despite ANNs' many attractive features, actually constructing, training and using an ANN is far from straightforward, and a good understanding of the underlying theory of neural networks is essential.

In any event, the ANN discussed in the fitting experiments that follow is a sophisticated lightcurve-fitter, developed by Vermaak (2003, 2007) specifically to fit SBLM lightcurves. Unlike the other algorithms discussed here, Vermaak's ANN-based fitter does not perform any iterative search through parameter space; instead, the mapping from lightcurves to model parameters is approximated directly by a complicated function derived from a very large set of training lightcurves. Although the ANN does require several hours to train, once-trained, fits can be generated very rapidly.

In many senses, the philosophies underpinning the ANN-based fitter and the EA are antithetical: the neural network is highly optimised for the fitting of SBLM lightcurves – indeed, it is a truly bespoke algorithm – but, unless it is retrained from scratch, it can *only* fit SBLM lightcurves, and it certainly cannot handle more general optimisation

problems. As with the EMMA, however, the ANN's output can readily be fine-tuned by a dedicated local-optimiser; accordingly, the ANN's performance both with and without the benefit of amoeba optimisation is considered in the fitting experiments that follow.

5.2 Fitting experiment setup

The primary aim of the fitting experiments was to study the feasibility of using an EA (viz., EMMA) to explore binary model parameter spaces. In order to quantify EMMA's performance in both an absolute and a relative sense, it was pitted against the other optimisation algorithms described in Sec. 5.1, and given the task of fitting several hundred simulated binary-lens lightcurves, each containing between one and four peaks. SBLM lightcurves with more than four peaks are theoretically possible though, assuming the parameter ranges in Table 5.1, very rare.

Following Vermaak (2003), the SBLM with parameter ranges in Table 5.1 (these ranges cover most of the so-called 'lensing zone', and thus the geometries with the highest likelihood of detecting secondary lenses) was used to generate lightcurves comprising 100 magnification points, starting at a random time $t_{\rm start}$, where

$$-3 < \frac{t_{\rm start} - t_{\rm m}}{t_{\rm E}} < -2, \tag{5.1}$$

and ending at a random time $t_{\rm end}$, where

$$2 < \frac{t_{\rm end} - t_{\rm m}}{t_{\rm E}} < 3. \tag{5.2}$$

Note that since $f \rightarrow 1$ (cf. Table 5.1 and Eqn. 2.13), the source is assumed to be unblended.

The use of synthetic data, where input parameters are known exactly, meant that fitting success could be gauged not only in terms of goodness-of-fit, but more importantly, also in terms of adequate sampling of the 'correct' regions of parameter space. Accordingly, a fitting sequence was deemed successful if it returned at least one model satisfying $\Delta\chi^2/\nu < 1$, with errors in all fitted parameters of less than 10 percent; a weaker criterion of $\Delta\chi^2/\nu < 1$ only, corresponding to a good fit(s) but not necessarily to closeness in parameter space, was also considered. Here, χ^2 is the chi-square goodness-of-fit statistic for a fit, defined thus:

$$\chi^2(\vec{p}) := \sum_t \frac{(m(t) - \hat{m}(t\,|\,\vec{p}))^2}{\sigma_t^2}, \tag{5.3}$$

where $m(t)$ is the t-th simulated observation (magnitude), σ_t^2 is the variance of the error distribution assumed for that observation, and $\hat{m}(t\,|\,\vec{p})$ is the magnitude predicted by the fitted model, defined by the parameter vector $\vec{p}$, also at time t. The 'delta chi-square' statistic, $\Delta\chi^2/\nu$, was defined as:

$$\Delta\chi^2(\vec{p})/\nu := (\chi^2(\vec{p}) - \chi^2_{\rm true})/\nu, \tag{5.4}$$

Parameter	Units	Minimum	Maximum
a	–	0.6	1.7
b	–	0.001	1
m_0	mag	18	22
q	–	0.1	1
t_{E}	d	5	50
t_{m}	d	-20	20
θ	rad	0	2π
f	-	1.00	1.00

Table 5.1 – Allowed parameter ranges for all fits and simulated lightcurves in this chapter.

where χ^2_{true} is the chi-square statistic for the *input* lightcurve (sans observation errors), and ν is the number of degrees of freedom associated with the fitted model in question.

The fits performed by the grid search, the iterated simplex and the evolutionary algorithm were all guided directly by the minimisation of a chi-square statistic (viz. $\chi^2(\vec{p})$, as defined in Eqn. 5.3; $\Delta\chi^2(\vec{p})/\nu$ was calculated only *after* fits were completed). Although the ANN-based fitting did not involve the direct computation of any chi-square statistic, or indeed any goodness-of-fit statistic,[2] its *training* was based on a chi-square metric, so in the context of these fitting experiments, the ANN might be thought of as an indirect means of attempting a chi-square minimisation.

The fitting experiments were carried out on a fairly high-end (by 2012 standards) workstation with a six-core, 12-threaded, 3.6 GHz processor, and 6 GB of 1333 MHz ECC RAM. To level the playing fields, all algorithms except for the ANN were parallelised, given an identical function to map trial solutions to lightcurves (see Sec. 5.6.6), and were restricted to 20 minutes per fit (the ANN generated fits far more rapidly, and in any case, would not have benefited from a longer run time; once it had generated a solution, it was 'done'). In the case of the grid search algorithm, parallelisation entailed using all the available processor threads to evaluate twelve grid-points concurrently, and in the case of the iterated simplex algorithm, twelve downhill-simplex runs were carried out in parallel. The parallelisation of EMMA has already been discussed in Sec. 4.3.2.

It should be noted that none of the algorithms tested here explicitly incorporated any information specific to the microlensing modelling problem. This is discussed in more detail in Sec. 5.6.2.

[2]The term 'badness-of-fit' would be more appropriate, since, all else being equal, a larger χ^2 statistic translates into a poorer fit

Method	Reference	Percentage successful fits			
		1 peak	2 peaks	3 peaks	4 peaks
Grid search	This work	1 (1)	0 (0)	0 (0)	0 (0)
Iterated simplex	This work	43 (98)	36 (93)	26 (78)	16 (50)
ANN by itself	Vermaak	4 (6)	0 (0)	0 (0)	0 (0)
ANN with amoeba	Vermaak	70 (96)	70 (82)	62 (72)	68 (76)
EMMA by itself	This work	93 (100)	96 (98)	90 (93)	89 (89)
EMMA with amoeba	This work	97 (100)	98 (100)	97 (98)	94 (94)

Table 5.2 – Comparison of several techniques used to fit randomly-generated, noise-free binary lightcurves. A fitted model was deemed successful if it had $\Delta\chi^2/\nu < 1$ *and* errors in all fitted parameters $< 10\%$; success rates using the weaker criterion of $\Delta\chi^2/\nu < 1$ only are shown in parenthesis.

5.3 Results: noise-free lightcurves

In the first experiment, lightcurves were uniformly-sampled and noise-free; again for consistency with the formalism of Vermaak (2003), a value of $\sigma_t = 0.01\ \forall\ t$ was adopted[3] for the purposes of calculating $\Delta\chi^2/\nu$. Although this $\Delta\chi^2/\nu < 1$ criterion had little statistical value in its own right, it did at least provide a means to quantify how closely a fitted lightcurve matched an input one (since, in the noise-free case, $\chi^2_{\rm true} = 0$ and $\Delta\chi^2 = \chi^2$, which is in turn proportional to the mean squared error for the fit), and thereby to rank the different algorithms.

Results from this test are presented in Table 5.2.

Sans the benefit of the amoeba-based local optimisation, the accuracy of the ANN-based fits was unacceptably poor (success rate close to 0%); the same applied to the fits obtained via the grid search. The iterated simplex approach was somewhat more profitable, yielding a success rate of about 30%. More impressively, the addition of the local optimisation improved the fits of the ANN considerably, and when viewed in terms of success rates (as high as 70% for the ANN), the results seem quite respectable. Still, for critical work, a failure rate of around 30% is far from ideal.

Most impressive was EMMA which, coupled with the amoeba method, was easily the most accurate of all the algorithms: it failed to yield a very accurate fitted model for fewer than 4% of the lightcurves. Even without the additional amoeba-based optimisation, EMMA performed extremely well in its own right, which bears testimony to the algorithm's capabilities on both the global and local optimisation fronts. Additionally, in the handful of cases where EMMA (with or without the amoeba fine-tuning) failed, allowing the algorithm to run a little longer (e.g. for 30 minutes instead of the arbitrarily-imposed 20 minute limit) always yielded successful fits. The same concession applied to the grid search and the iterated simplex algorithm, but the time required to ensure successful fits was invariably much longer.

[3] Roughly speaking, this scaling – though otherwise arbitrary – meant that fits satisfying the $\Delta\chi^2/\nu < 1$ criterion looked good 'by eye'.

Fig. 5.1 gives examples of some typical fits performed by EMMA.

All algorithms fared better when fitting simpler (single- or double-peaked) lightcurves than when fitting the more complex ones, although for EMMA this difference was very small. Indeed it is quite surprising just how accurately the algorithm managed to home in on model parameters – in the case of successful fits, almost always to an accuracy of much better than 1% on all parameters, even when the lightcurves had little apparent structure.

5.4 Development of a simple noise model

The lightcurves used in the fitting experiments described in the previous section were noise-free, meaning that it was possible, in principle at least, to obtain perfect fits to the simulated lightcurves (and if it were not for parameter degeneracies and ambiguities, there would be a $1:1$ correspondence between perfect fits and *correct* models/solutions). This is not at all representative of real-world microlensing events, where photometric noise is always present to some extent – and unfortunately, the presence of photometric noise always has a negative impact on one's prospects of obtaining a good fit, and of accurately extracting parameters from a lightcurve (Schneider et al., 2006).

Accordingly, a simple model of photometric noise was developed, based on fits to 1000 microlensing events observed during the 2011 campaign of the Optical Gravitational Lensing Experiment (Udalski, 2003); the full dataset comprised approximately 1.5 million I-band magnitudes, along with estimated photometric errors, seeing estimations and sky levels.[4]

In general, the photometric noise associated with real microlensing events will depend on data reduction methods, field crowding, source brightness, and a number of other observational factors. Fortunately, the OGLE dataset suggested a very strong correlation between source magnitude and photometric noise (rank correlation coefficient 97%), indicating that – to an excellent first approximation – photometric noise can be modelled as a function only of source magnitude, with all other factors relegated to second-order corrections to the basic model.

Linear regression led to the following best-fitting model for photometric noise, as a function of I-band magnitude (based on data in the range $14 \lesssim m_I \lesssim 22$):

$$\log_{10}\sigma_I = 0.3416^{0.3418}_{0.3414} \cdot m_I - 7.7095^{7.7132}_{7.7059}, \tag{5.5}$$

where σ_I is to be interpreted as the standard deviation of a Gaussian distribution of observed magnitudes around some unobserved true magnitude, m_I; the superscripted and subscripted numbers indicate, respectively, the upper and lower 3σ confidence bounds on the model parameters. This model, as well as the data on which it was based, is illustrated in Fig. 5.2.

This model seems rather optimistic for bright sources ($m_I \sim 14$), and even the assumption of Gaussian noise is probably naïve (Dominik, 2008); still, the model does at

[4]The data are available online at `http://ogle.astrouw.edu.pl`.

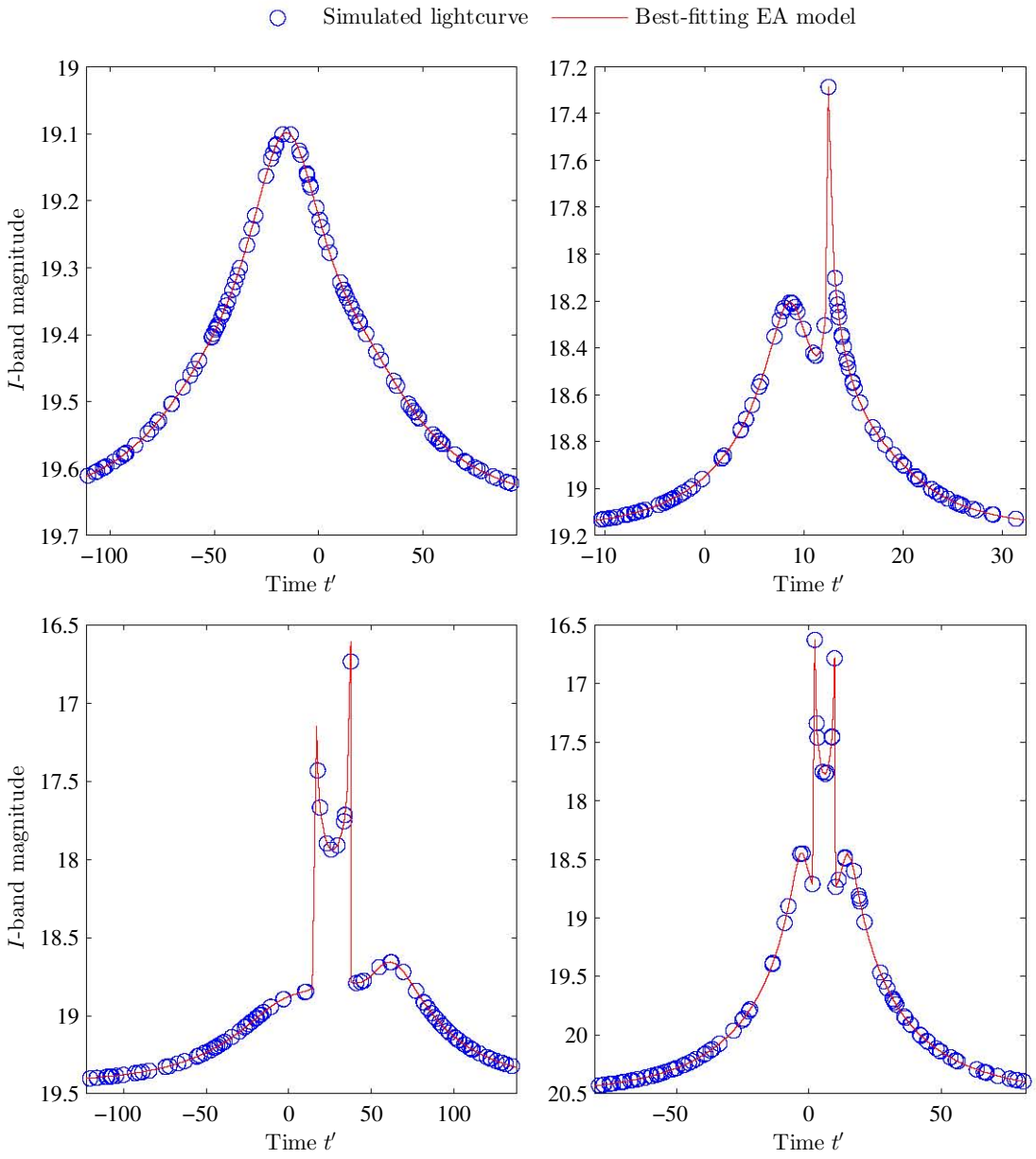

Figure 5.1 – Typical fits performed by EMMA in the case of noise-free lightcurves. Fits to lightcurves containing 1, 2, 3 and 4 peaks are shown. Note: on account of limited space, the labels for the vertical axes in the right-hand panels have been omitted; the implied labels for these axes are identical to the labels in the left-hand panels.

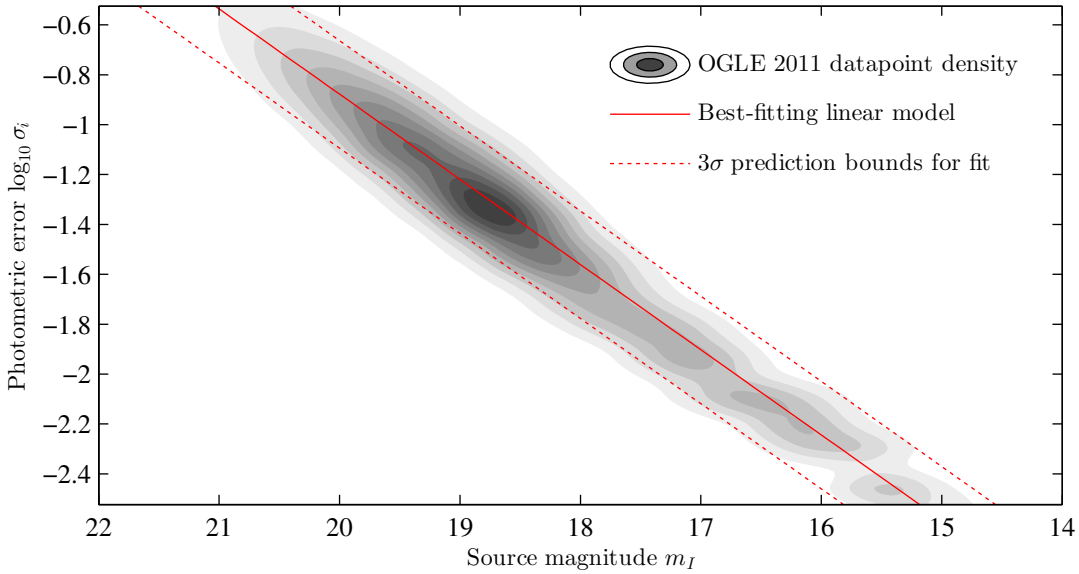

Figure 5.2 – The best-fitting model for photometric noise vs. I-band magnitude, along with a representation of the original dataset on which the model was based. The filled/shaded contours indicate data density, with darker contours representing proportionately greater data densities (data density is plotted because it was neither feasible, nor desirable, to plot all ~ 1.5 billion of the original datapoints).

least provide a means of making the simulated lightcurves used for fitting experiments somewhat more realistic and more challenging to fit. This noise model is adopted in the fitting experiments in the next section.

5.5 Results: noisy lightcurves

In order to make the fitting problem more challenging, the following changes were made to the basic experimental setup:

(i) temporal sampling of lightcurves was randomised by drawing observation times from a rectangular distribution on $(t_{\rm start}, t_{\rm end})$, thus allowing for gaps, corresponding e.g. to bad weather, in the data; and

(ii) Gaussian noise, as per the best-fitting model defined by Eqn. 5.5, was added to all datapoints.

For consistency's sake, all other aspects of the experiment were left unchanged.

Table 5.3 gives an indication of how EMMA, the iterated simplex algorithm and the grid search algorithm fared on this problem. Once again the evolutionary algorithm outperformed the other algorithms; the failure rate of EMMA, with amoeba fine-tuning, was

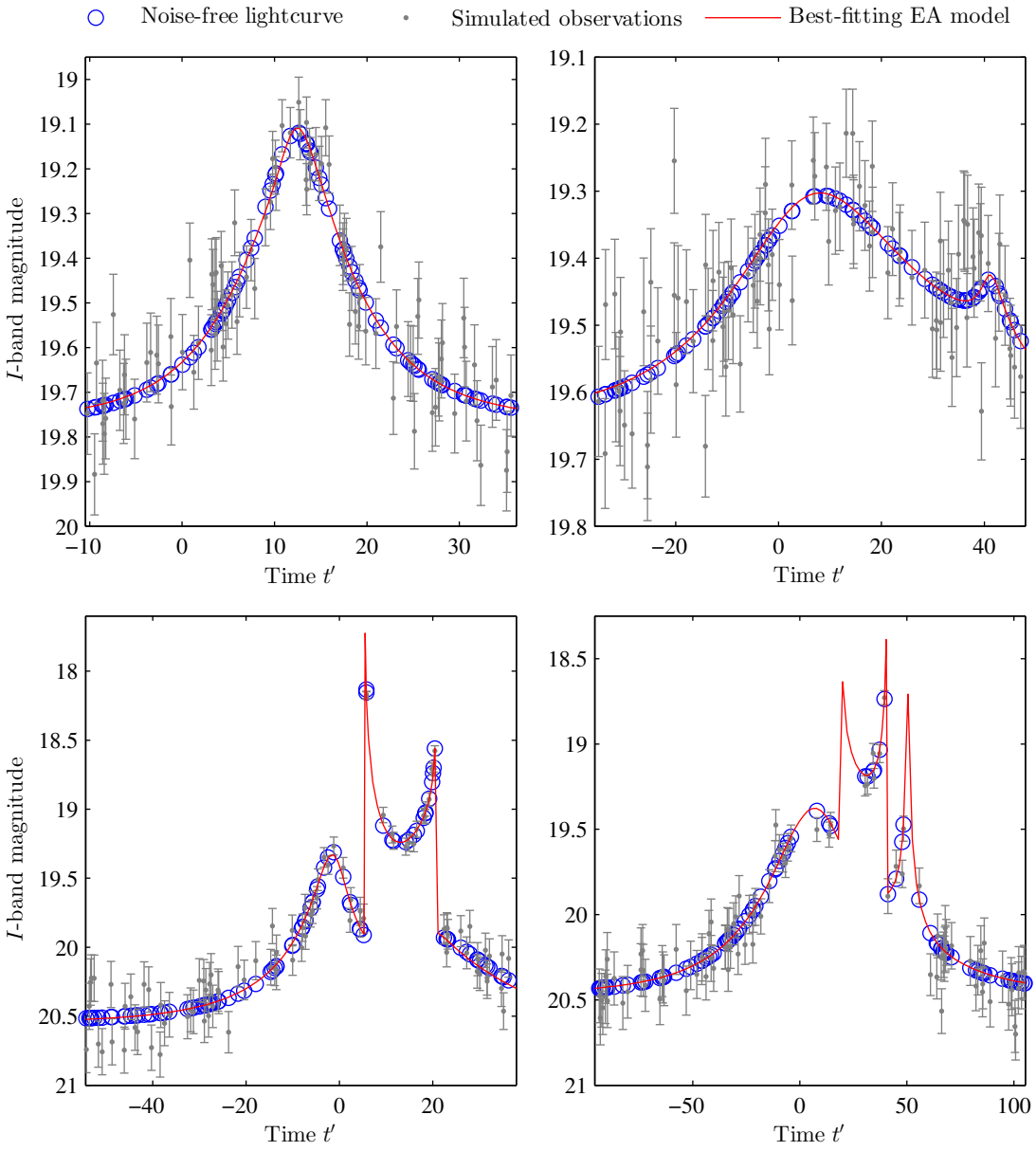

Figure 5.3 – Typical fits performed by EMMA in the case of noisy lightcurves. Fits to lightcurves containing 1, 2, 3 and 4 peaks are shown. All fits were performed to the *noisy* data; the noise-free, input data are shown only for the sake of comparison. Note: on account of limited space, the labels for the vertical axes in the right-hand panels have been omitted; the implied labels for these axes are identical to the labels in the left-hand panels.

Method	Reference	Percentage successful fits			
		1 peak	2 peaks	3 peaks	4 peaks
Grid search	This work	1 (26)	0 (17)	1 (1)	1 (2)
Iterated simplex	This work	44 (92)	30 (92)	14 (52)	16 (44)
EMMA by itself	This work	84 (100)	83 (84)	77 (78)	94 (100)
EMMA with amoeba	This work	97 (100)	95 (99)	92 (100)	94 (100)

Table 5.3 – Comparison of several techniques used to fit randomly-generated lightcurves, including photometric noise, and randomised temporal sampling. Success criteria are as in Table 5.2.

only around 5%, which compares very favourably with the iterated simplex algorithm's failure rate of nearly 75%. The success rate of the grid search algorithm actually seems to improve on this more challenging problem – at least, when using the weaker success criterion of $\Delta\chi^2/\nu < 1$ only – but this is easily explained in terms of the arbitrary scaling of σ_t^2 used in the noise-free case.

Unfortunately, no data were available to facilitate a direct comparison, i.e. using an identical noise model and randomised temporal sampling, with Vermaak's ANN; his conclusion, however, based on experiments using a broadly-similar noise model, was that noise has detrimental effects on the accuracy of ANN-based fitting, but that an ANN can at least be designed/retrained to mitigate these effects (Vermaak, 2007). As such, even if one assumes that the ANN can perform as well with the noisy lightcurves as it did with the noise-free ones, EMMA still emerges the far more accurate algorithm.

Fig. 5.3 gives examples of some typical, noisy-lightcurve fits performed by the EMMA.

5.6 Discussion

5.6.1 Advantages of the EA

While EMMA takes longer than the ANN to generate fits – once suitably trained, the ANN can perform a fit in under a minute! – it certainly offers a search efficiency far beyond that of mere brute-force approaches, as evidenced by its clear outperformance of the grid search and even iterated simplex approaches, both of which were calibrated to have the same computational footprint as the EA. The computational expense notwithstanding – and at any rate, a fitting time of several minutes would be more than fast enough for modelling ongoing events – EMMA offers a number of striking advantages over even the bespoke, ANN-based approach to lightcurve fitting:

(i) the fitting accuracy of the EMMA is superior to that of the ANN;

(ii) the algorithm itself is much simpler (and, speaking more generally, it is usually easy to code from scratch and to configure an EA, which is certainly not the case with an ANN);

(iii) the algorithm can be used straight 'out of the box', and requires no training;

(iv) the algorithm is far more robust because it does not need to be retrained/reconfigured to cope with noise, extensions to the basic lightcurve model, or even entirely different models; and

(v) the algorithm does not yield a single, 'all or nothing' fit – a single run yields very many fitted models, and the longer the algorithm is allowed to run, the better the chance of pinpointing the globally optimal models(s).

5.6.2 Incorporation of problem-specific information

As already noted, none of the algorithms tested incorporated any information specific to the structure of the microlensing modelling problem, e.g. feature-based parameterisations, knowledge of lightcurve morphologies, knowledge of the topologies of underlying caustic/critical curves and source trajectories, etc. Incorporating such prior knowledge could be used to make most of the algorithms, except for the ANN, more efficient: that is, rather than letting them search 'blindly' through the full SBLM parameter space, problem-specific information could be used to reduce the volume of the space of feasible parameters, and could provide clues as to where good solutions are likely to be found (Kains et al., 2009). Insofar as this work goes, the main reason microlensing-specific information was not incorporated was to facilitate a more fair comparison of the other algorithms with Vermaak's ANN, into which the incorporation of problem-specific information would have been very difficult. Still, some more general arguments could be made in favour of the non-inclusion of such information.

Incorporating knowledge of the SBLM and its associated lightcurves, for example, would increase the complexity and reduce the generality of the algorithms, and new information would need to be incorporated manually if one wanted to fit, say, a triple-lens model instead of a binary model. Indeed, if the computational power is available for an algorithm to perform fits in a reasonable amount of time, the manual incorporation of problem-specific information would be, if nothing else, an unnecessary expense of human effort.

5.6.3 The importance of parallelisation

A fitting time on the order of several minutes does sound appealing, though unfortunately the simple binary-lens model adopted here neglects many effects usually associated with real-world microlensing events, including blending,[5] parallax, instrumental and finite-source effects. Incorporating some of these effects (e.g. blending) poses no significant difficulties to modellers, and requires only simple tweaks to the basic SBLM; computing the magnification of a finite source star, however, can take around two orders of magnitude longer than the corresponding calculations under the point-source approximation

[5] Fitting experiments akin to those in Sec. 5.5, but allowing for significant blending, are presented by Rajpaul (2012). With the addition of blended light, EMMA's success rates remain comfortably above 90%.

(Vermaak, 2007), and unfortunately, finite-source effects generally cannot be ignored when dealing with planetary signals (Vermaak, 2000). This being the case, it might seem that the claims about the evolutionary algorithm's speed would be invalidated when it comes to real-world fitting problems. To make matters worse, the cadence of modern wide-field surveys tends to be up to two orders of magnitude higher than in the lightcurves simulated here (Shvartzvald & Maoz, 2012) – this,[6] combined with having to deal with finite-source effects, would suggest an increase in fitting times by about four orders of magnitude!

Certainly, some performance gains can be expected from a meta-optimisation of EMMA's design and/or control parameters (i.e. 'fine-tuning' the algorithm specifically for fitting microlensing lightcurves); further gains could be expected by incorporating problem-specific information into the algorithm, as suggested above.

Far most importantly, however – as noted in Sec. 3.6 – EAs are highly amenable to parallel implementations, and speedups of up to four orders of magnitude, using only quite modest GPU hardware, have been demonstrated in the literature. Therefore, if one is willing to invest some time to port an evolutionary algorithm to a GPU platform, it seems that it would certainly be possible to constrain the algorithm's fitting time to the order of several minutes (rather than days or weeks), even when dealing with very complex lightcurves. This might not seem much better than existing state-of-the-art algorithms (at least one of which seems to be able to perform complex fits in the ballpark of an hour: see Bozza, 2012) but it does illustrate the general point that the use of evolutionary algorithms – which tend to be versatile, robust, conceptually simple, etc. – represents a viable alternative to developing sophisticated, bespoke algorithms in order to solve difficult modelling problems.

Incidentally, despite all the reservations about the SBLM used in the fitting experiments, Vermaak (2007) showed that this model *could* be used to perform useful fits even to lightcurves including significant finite-source effects, the *proviso* being that the very localised parts of the lightcurves (e.g. the tops of peaks) in conflict with the SBLM's assumption of a point-like source simply be excluded from the fitting. Given, then, that the potential for using the SBLM to fit real-world binary events has already been demonstrated in the literature, and also for brevity's sake, examples of EA-based SBLM fits to real-world lightcurves are not presented here. The next chapter of this dissertation does, however, present examples of EMMA not only performing fits to, but also making predictions from, real-world data.

5.6.4 Dynamics of the evolutionary search

It is instructive to study the topographies of some representative regions of the search spaces associated with binary fitting problems, as well as the 'trajectories' taken by evolutionary populations within these spaces: the former in order better to appreciate the many difficulties of the optimisation problem associated with fitting binary lightcurves,

[6] Current work does, however, point to the feasibility of using data compression to reduce full lightcurves to a smaller number of information-carrying datapoints, in order to accelerate analysis of binary events (Heavens et al., 2000; Hundertmark, 2012).

and the latter in order to get a better feel for the dynamics and operation of EMMA on such problems.

Accordingly, Fig. 5.4 provides a visualisation of some of the optimisation spaces – or at least, 2D slices thereof – associated with typical, noisy binary lightcurves. The plots were generated from randomly-selected lightcurves used in the experiments in Sec. 5.5. Pathologies including degeneracy, low contrast, multiple optima separated by steep walls, and non-smoothness can be seen in these plots, and it should be clear why a conventional (e.g. gradient-based) algorithm would have little hope of locating solutions. Even Charbonneau's optimisation problem (see Eqn. 3.3) looks quite tame by comparison! It

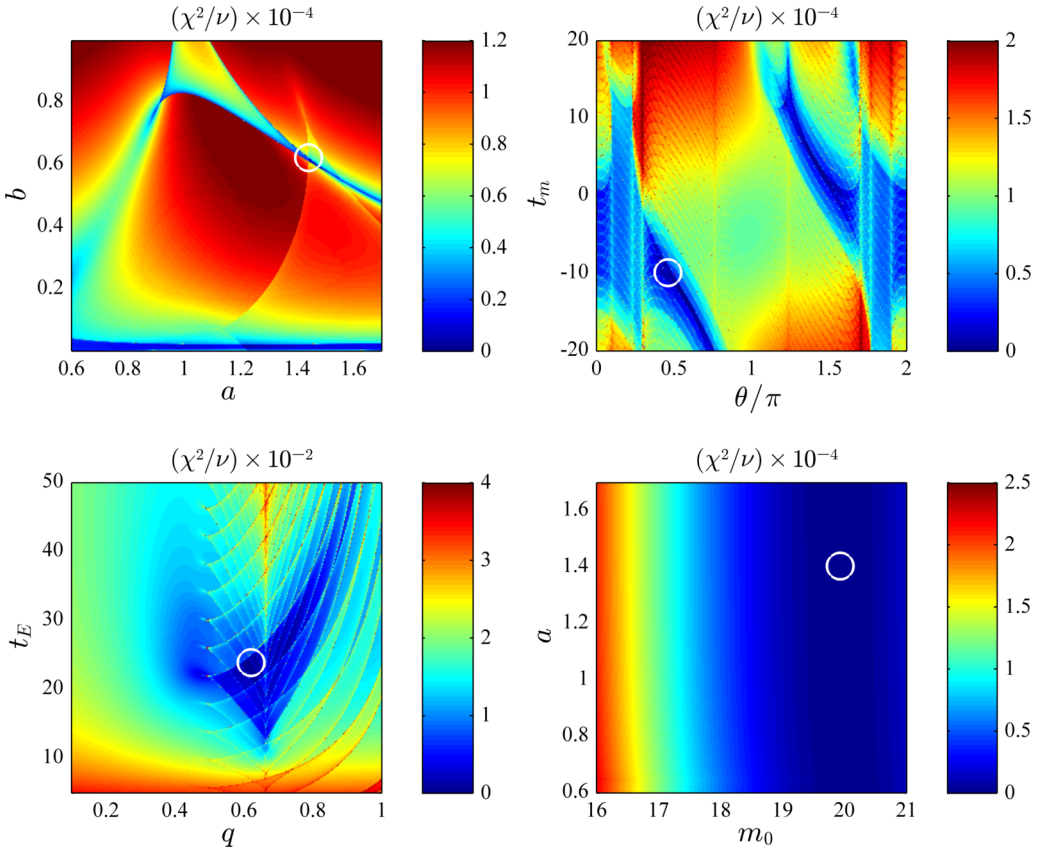

Figure 5.4 – A visualisation of some of the pathologies inherent in the optimisation landscapes associated with noisy binary lightcurves. In each panel, fitness is plotted as a function of two free parameters, with other parameters fixed at their 'correct' values, i.e. the values used to generate the lightcurves being fitted. The locations of the correct values for the two free parameters are indicated by white circles. Starting from the lower left panel and moving clockwise, we see evidence for multiple minima separated by steep, narrow walls; a deceptive landscape where moving downhill usually entails moving *away* from the global optimum, which has a very small basin of attraction; evidence of parameter degeneracy; and an optimum surrounded by a region of very low local contrast.

is interesting to try to reconcile EMMA's exploration mechanisms – driven by solution recombination, jump and creep mutation etc. – with the nightmarish topographies of the search spaces with which they are confronted. One can quite easily imagine, for example, an opportune creep mutation neatly boosting a trial solution over a steep, narrow wall

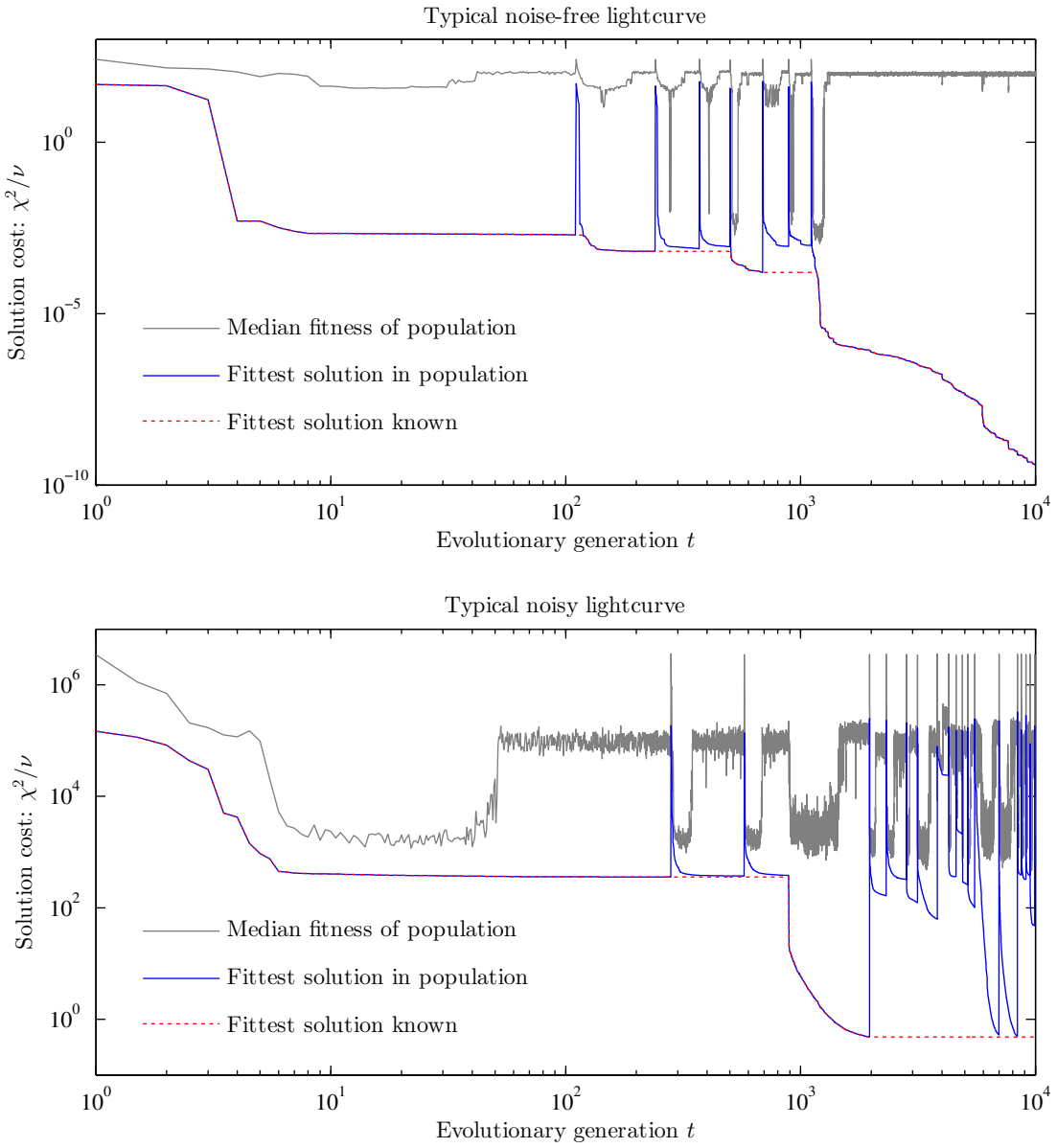

Figure 5.5 – A visualisation of two typical trajectories taken by an evolutionary population, one in the case of performing a fit to a noise-free lightcurve (upper panel), and the other in the case of a fit to a noisy lightcurve (lower panel). Note the logarithmic horizontal axes.

in the fitness landscape.

Perhaps more informative are plots that illustrate the changing fitness of an evolutionary population, as it explores the search space and homes in on good solutions. Fig. 5.5 presents two such plots, one for a noise-free lightcurve, and one for a noisy lightcurve.

In the noise-free case, we see that the evolutionary population typically stagnates a few times around different local optima, and after each such stagnation, the population is reinitialised. Before long, though, the vicinity of the global optimum is sampled, and from there on in, the evolution of the fittest solution in the population becomes a process of local optimisation (of course, if one knew beforehand that one had already homed in on the global optimum, the whole evolutionary sequence could be terminated, and a dedicated local optimiser allowed to take over).

Notice that the *median* population fitness tends to increase when rapid improvements are being made to the fittest solution in the population, and to deteriorate when the fittest solution in the population stops improving – this is a consequence of the dynamical adjustment of mutation rates, described in Sec. 4.1, which serves to shift emphasis to or away from global exploration. For example, in the latter stages of the evolution, i.e. after generation $\sim 10^3$, an excellent solution – in fact, the global optimum – has already been located, but the jump-mutation rate is kept high to provide a form of insurance against having missed the global optimum, and then wasting all one's time exploiting an apparently good though actually suboptimal solution.

The dynamics of the evolution are even more interesting in the noisy case, where the global optimum now corresponds to $\chi^2/\nu \sim 1$, rather than $\chi^2/\nu = 0$. In the illustrated example, the population stagnates a couple of times around incorrect optima, until the correct solution is located at around generation 10^3. However, the algorithm does not *know* that this is the correct solution, so the evolution is restarted, and the population is allowed to converge to a number of other potential solutions. In fact, the global optimum is independently re-sampled two further times during the full evolutionary sequence, and though the fittest solution in the final generation of the evolution does not correspond to the globally optimal solution, the global optimum does correspond to the fittest solution sampled over *all* t.

5.6.5 On Monte Carlo methods

It is worth mentioning – even if only in passing – that Monte Carlo methods are ubiquitous in the field of microlensing modelling (Sumi et al., 2010; Shin et al., 2011; Skowron et al., 2011). During the course of this work, a Markov chain Monte Carlo (MCMC) method, incorporating an adaptive Metropolis sampler and a delayed rejection mechanism (Haario et al., 2006), was deployed and used to perform fits on single-lens events. The MCMC method was found to be nearly two orders of magnitude slower than EMMA, and so no in-depth tests of the MCMC method were carried out for binary-lens events. In any event, even if an EA is able to perform fits more quickly, Monte Carlo methods would remain useful e.g. for estimating uncertainties in fitted parameters found by the EA (Press et al., 2007; Charbonneau, 2002a).

5.6.6 On solving the forward problem

The discussion in this chapter would not be complete without a few remarks about the function used to map SBLM parameters to lightcurves, i.e. to solve the forward problem associated with the inverse problem of extracting parameters from lightcurves: after all, it was this external function that EMMA, the grid search algorithm, and the iterated search algorithm all called, countless millions of times (in order to map trial parameters to a measure of solution fitness), when fitting lightcurves!

The 'first attempt' implementation of this mapping function was coded in Python, but in the interest of speeding up the code, a number of changes were made to the original implementation. The major changes were, in order from first to last, the following:

(i) the choice of algorithm used to find the roots of the lensing polynomial (see Eqn. 2.15) was optimised;

(ii) careful factorisation was used to optimise the computation of coefficients of the lensing polynomial (see Eqns. 2.16 to 2.21);

(iii) the code for the polynomial root-finding algorithm was pared down and optimised;

(iv) the entire parameter-to-lightcurve-mapping function, incorporating the aforesaid improvements, was ported from Python to MATLAB;

(v) the function's structure was adapted to take advantage of MATLAB's array-based arithmetic; and lastly,

(vi) the final MATLAB implementation was ported directly to C++.

To quantify the effects of the aforementioned incremental improvements,[7] a platform-independent measure of a code's speed needed to be devised (for example, it'd be very well to say a code takes 10 minutes to generate a thousand, 100-point lightcurves, using a computer with dual 3 GHz processors; but how would this speed translate when generating 500-point lightcurves on a quad-core, 2 GHz laptop?). Accordingly, code speed was normalised to remove the effects of different processor clock speeds (measured in GHz), different numbers of available processor cores (assuming 100% parallelisation efficiency), and lightcurves of different lengths. Thus, a code's speed was quantified in terms of the number of SBLM lightcurve 'points per GHz per processor core per second' (points/GHz $\cdot$ core $\cdot$ s, hereafter 'pGcs') it was capable evaluating.[8]

For example, if a code's speed is listed as 100 pGcs, this means that using a single-core, 1 GHz processor, the code should be able to evaluate ~ 100 amplification points in one second; on a 2 GHz, quad-core system, it should be able to evaluate ~ 800 points in one second; and so on.

[7] Thorough debugging and verification – which entailed testing code on problems with 'known' answers, testing code on different platforms, etc. – was performed throughout the course of the development of this and related code.

[8] Increasing the size of the CPU cache was found to have little effect on the speed of the code, so cache size was not included in the normalisation.

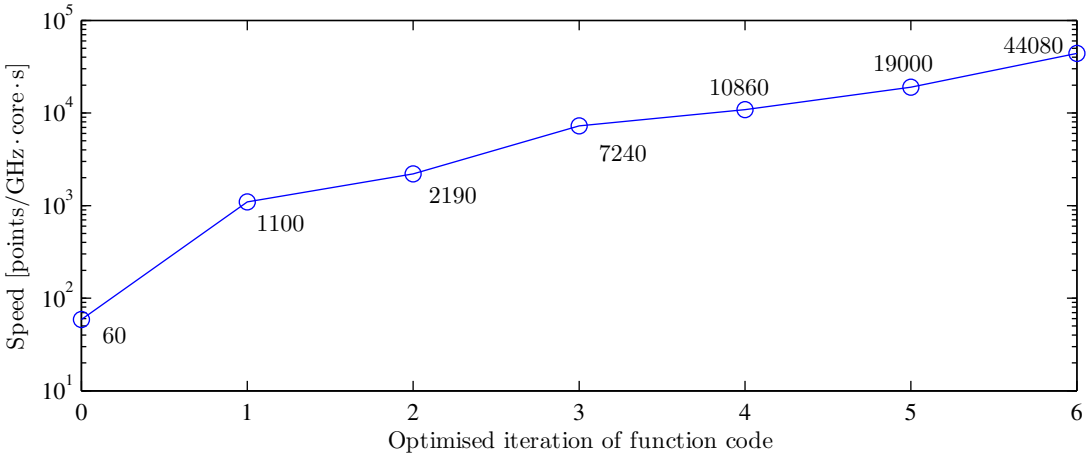

Figure 5.6 – Illustration of the incremental but ultimately very significant speedups made to the code used to map SBLM parameters to lightcurves. The numbers on the horizontal axis indicate the sequential improvements listed in the discussion in Sec. 5.6.6.

The effects of the six aforementioned code improvements are quantified (in terms of pGcs) and illustrated in Fig. 5.6. Whereas the initial implementation of the function had a speed of only around 60 pGcs, the final implementation was no less accurate yet $\sim 750\times$ faster, with a speed of around 44000 pGcs.[9]

As may be inferred from Fig. 5.6, the single biggest speedup ($\sim 20\times$) came simply from making a careful choice of root-finding algorithm. A number of root-finding algorithms – including Laguerre's method, Halley's method, and the Durand-Kerner method (Ralston & Rabinowitz, 2001) – were tested; ultimately, an industry-standard implementation of a modern method (Edelman & Murakami, 1995), based on efficiently computing the eigenvalues of a polynomial's companion matrix, was chosen, both for its speed and accuracy.

Further improvements to the code are almost certainly possible (not least because almost all code was written from scratch, rather than having been adapted from existing libraries) – at any rate, the important point illustrated here is that enormous performance gains can often be obtained simply by ensuring general coding and algorithmic efficiency, regardless of the optimisation technique used to fit lightcurves.

5.7 Conclusions

It was demonstrated in this key chapter that evolutionary algorithms seem well-suited to fitting binary-lens microlensing lightcurves. Despite the difficulty of this fitting problem,

[9] The absolute speeds in Fig. 5.6 should be regarded as optimistic: such speeds will are only obtainable under ideal testing conditions, where 100% of a computer's processing resources are made available to the code, all lightcurve points are computed without any interruptions, parallelisation overheads are negligible, etc. The relative speeds of the different code iterations, however, ought to be invariant.

EMMA, an evolutionary algorithm, yielded excellent fitting accuracy whilst maintaining a relatively modest computational footprint, and also offering a number of other desirable properties (robustness; versatility; easy parallelisation; etc.). As such, the results presented here provided proof of concept for using an EA as the basis for real-time, autonomous modelling of microlensing events.

Although EMMA's performance certainly is promising, it does however remain to be seen how the algorithm will fare when faced with more difficult microlensing modelling problems. Indeed, the simple binary-lens model and simulation parameters adopted in the fitting experiments (primarily for the sake of allowing a direct comparison with the results of Vermaak) led to the exclusion of many effects usually associated with real-world microlensing events, including blending, finite-source, parallax and instrumental effects. Some of the parameter ranges used here would need to be extended – or, better yet, replaced with more realistic Bayesian priors that avoid hard limits – to cover cases where the secondary lens is an exoplanet. Finally, to be more representative of modern, high-cadence surveys, temporal sampling of simulated lightcurves would need to be increased by between one and two orders of magnitude (Shvartzvald & Maoz, 2012) – this would increase fitting times, but should also help to resolve some model ambiguities.

Still, since EMMA managed to move effortlessly from single-lens (during its development) to binary-lens (Sec. 5.3) to noisy binary fitting problems (Sec. 5.5), one can speculate that fitting problems involving more sophisticated microlensing models should not pose major obstacles to the algorithm. Of course, given a more sophisiticated model, one would need to sacrifice some fitting accuracy, and/or harness more computing power, as suggested in Sec. 5.6.3, if one wanted to leave fitting times unchanged.

6

EA-based microlensing forecasts

This chapter presents fitting experiments that were set up and run to provide proof of concept for using evolutionary algorithms to model *active* (ongoing) microlensing events. At the same time, the fitting experiments are used to quantify, in broad terms, the usefulness of performing real-time fits to incomplete binary-lens lightcurves, in order to provide forecasts of possible upcoming features in the lightcurves.

Sec. 6.1 of this chapter provides an introduction, and sketches the need for real-time modelling of active microlensing events. Sec. 6.2 describes the relevant fitting experiments, Sec. 6.3 presents the results of these experiments, and Sec. 6.4 contains a discussion of the results presented in the preceding section. Sec. 6.5 provides a few examples of the EA-based modelling/forecasting approach applied to real-world data, and finally, Sec. 6.6 concludes.

6.1 Introduction

The previous chapter of this dissertation demonstrated that evolutionary algorithms appear to be well-suited to modelling microlensing lightcurves, and it provided some proof of concept for the use of EAs as the basis for real-time, autonomous modelling of microlensing events.

On the one hand, the notion of 'real-time modelling' is rendered almost irrelevant when one is dealing with *completed* microlensing events (and of course, all of the lightcurves considered in the previous chapter were simulates of completed events): it would be senseless to rush to model a lightcurve in a matter of minutes or hours, thereby risking very inaccurate conclusions, if one could instead afford to spend weeks, say, performing extremely comprehensive and meticulous modelling of the data. On the other hand, however, when one is dealing with *active* microlensing events – i.e., events whose lightcurves are still 'incomplete' – rapid modelling is potentially advantageous.

Suppose, for example, that an active lensing event has an interesting feature – a caustic crossing, perhaps – coming up in a matter of days, or even hours. While this will often be unknowable to anyone making a casual inspection of the event's lightcurve, real-time modelling of the event could, in principle, facilitate a prediction/forecast of the upcoming feature, thereby helping to ensure that there is a telescope pointing at the source at the critical moment (easier said than done, especially since, at any moment, there will usually multiple active events competing for observational attention!). Sufficient observational

coverage of the feature could, in turn, greatly enhance the amount of useful information that will eventually be extractable from the final lightcurve (Gaudi & Gould, 1997; Gaudi & Sackett, 2000). Such forecasts would be especially valuable for planetary signals, which tend to last less than a day (Gaudi, 2012).

In recent years, such real-time modelling efforts have become increasingly commonplace (see e.g. Bozza, 2012; Han, 2012) – unfortunately though, while such modelling efforts have already started to bear fruit, there have been very few quantitative analyses of their forecasting accuracy. In fact, the only paper dedicated to such a topic seems to be that of Jaroszyński & Mao (2001), and their study was limited to the analysis of just two binary lightcurves.

Accordingly, this chapter presents fitting experiments that were set up and run to quantify the potential for making real-time fits to incomplete binary lightcurves, in order to forecast upcoming features in the lightcurves. Of course, the algorithm used in these experiments was EMMA: since EMMA's speed and robustness have already been amply demonstrated (Chapter 5), this chapter will not be directly concerned with comparing EMMA's performance to that of other algorithms.

6.2 Fitting experiment setup

6.2.1 Overview of test setup

In brief, EMMA was given many (1000) different simulated binary lightcurves, each a truncated version of a complete lightcurve. Each complete lightcurve contained a feature that had not yet been 'observed' in the truncated lightcurve; EMMA's goal was to perform fits to the *truncated* lightcurve, and in so doing, to try to forecast the existence of, as well as the time of, the upcoming feature.

A simulated lightcurve was generated as follows. First, a random 'template' SBLM lightcurve, without any noise or missing data, was generated. Parameters for the lightcurve were drawn from a uniform distribution over the ranges in Table 6.1. The temporal sampling of the lightcurve was randomised by drawing 1000 different observation times from a rectangular distribution on $(t_{\rm start}, t_{\rm end})$, where

$$-3 < \frac{t_{\rm start} - t_{\rm m}}{t_{\rm E}} < -2, \tag{6.1}$$

and

$$2 < \frac{t_{\rm end} - t_{\rm m}}{t_{\rm E}} < 3. \tag{6.2}$$

This template lightcurve was then analysed to ascertain whether it was appropriate for inclusion in the fitting/forecasting experiments. The following three criteria all needed to be met in order for a lightcurve to qualify for inclusion in the experiments:

(i) the lightcurve needed to contain at least two peaks;

(ii) the time between the final and the penultimate peak had to be at least 1.0 days; and

Parameter	Units	Minimum	Maximum
a	–	0.5	5.0
b	–	0.001	2.0
m_0	mag	16	22
$\log_{10} q$	–	-5	0
$t_{\rm E}$	d	5	100
$t_{\rm m}$	d	-20	20
θ	rad	0	2π
f	-	0.01	1.00

Table 6.1 – Allowed parameter ranges for all fits and simulated lightcurves in this chapter. Note that the parameter ranges are expanded relative to those in Table 5.1.

(iii) the final peak had to be large enough to be detectable at 3σ above the noise level on either side of the peak.

If the template lightcurve did not satisfy *all three* of the aforementioned criteria, a new template lightcurve was generated and the criteria checked again. Once a suitable lightcurve was found, the template lightcurve was used to generate a set of simulated observations as follows:

(i) the template lightcurve was truncated at a random time, somewhere between the time of the penultimate peak and the inter-peak trough between the penultimate and the final peak;

(ii) the lightcurve was re-sampled by randomly adding points until the lightcurve once again comprised a total of 1000 'observations'; and

(iii) Gaussian noise, as per the best-fitting model defined by Eqn. 5.5, was added to all datapoints.

The simulated lightcurve thus constructed was then passed on to EMMA, whose goal was to forecast the structure of the lightcurve at future times, i.e. to try to reproduce the *template* lightcurve, based only on analysis of the noisy, *truncated* lightcurve. Importantly, the truncated lightcurve given to EMMA would not have contained any superficial clues as to whether a further peak(s) could be expected in the future.

A forecast made by EMMA[1] was deemed successful if it predicted the existence of the truncated peak *and* the time of that peak (i.e. the time of maximum amplification associated with that peak) to within a predefined tolerance or precision, $\Delta T_{\rm pred}$. $\Delta T_{\rm pred} = 0.5, 1.0$, and 2.0 d were considered. No penalties were imposed for an inaccurate prediction of the detailed morphology of a feature, or for predicting, in addition to the expected feature, spurious features in a lightcurve.

It should be emphasised that the simulated lightcurves used in these experiments differed in three key respects from those used in the previous chapter:

[1]Note: no local optimisation was used to refine EMMA's outputs.

(i) their temporal sampling was an order of magnitude higher, aligning the simulation more closely with modern, high-cadence surveys;

(ii) the ranges of the SBLM parameters used to generate lightcurves were expanded (cf. Table 5.1 and Table 6.1) – in particular, the expanded ranges made provision for low-mass exoplanets, as well as significant blending; and finally,

(iii) the lightcurves were no longer complete, i.e. they featured 'missing' data.

6.2.2 Motivation for test setup

The requirement that a lightcurve contain at least two peaks, i.e. that $N_{\text{peak}} \geq 2$, was enforced so that a lightcurve would always contain a well-defined feature for a fitting routine to forecast (viz., the final peak in the lightcurve), and moreover, so that even after this feature was wholly removed from the lightcurve, there would still remain some nontrivial structure in the lightcurve (i.e., at least one other peak) that a fitting routine might use to guide its search through parameter space. Expecting an accurate prediction of the existence and time of a peak, based only on a fit to a featureless slope, would be asking for a miracle!

Note that in a lightcurve, $\hat{m}(t \,|\, \vec{p})$, a peak was defined as a point $\hat{m}(t_i^* \,|\, \vec{p})$ such that:

$$t = t_i^* :\Rightarrow \frac{\partial \hat{m}}{\partial t} = 0, \frac{\partial^2 \hat{m}}{\partial t^2} > 0. \tag{6.3}$$

(Of course, since real lightcurves always comprise a finite number of points, in practice the derivatives in the definition need to be replaced with difference quotients.[2]) To put Eqn. 6.3 into words, a peak was defined as a local maximum in the trajectory of a source's amplification vs. time, or equivalently, a local minimum in the trajectory of its magnitude vs. time.

The requirement that the time between the final and the penultimate peak be *at least* 1.0 day, i.e. that $\Delta T_{\text{peak}} \geq 1.0$ d, where

$$\Delta T_{\text{peak}} := t^*_{N_{\text{peak}}} - t^*_{N_{\text{peak}}-1}, \tag{6.4}$$

was enforced to ensure that the two peaks could actually be resolved, given the likely temporal resolution of the simulated data. In fact, following the setup described in Sec. 6.2.1, any randomly-chosen observation will, with 99.7% probability, be separated from both the following and preceding observations by *less* than 1.0 d, with the median separation between all observations being about 1.5 h.

Finally, the requirement regarding the size of the final peak (3σ above the noise level) amounted to excluding from the experiments those lightcurves whose final peaks were so small that they would have been 'drowned out' by photometric noise (once such noise was

[2] Another technical point is that, under the point-source approximation, the derivatives in Eqn. 6.3 will be undefined at the exact time of a caustic crossing (in which case a peak could, instead, be defined as a point such that $\hat{m}(t) \geqslant \hat{m}(t_i^*) \,\forall\, t \neq t_i^*$). This consideration is not very important, however, when dealing with a lightcurve comprising a finite number of points.

added to the lightcurve), and would therefore have been very unlikely to be detectable at all, let alone possible to be forecast in advance.

The scheme used to truncate a lightcurve was chosen so that a truncated lightcurve would not contain any obvious indication that an *increase* in source magnification was imminent. The truncation time, $T_{\rm cut}$, was distributed as $T_{\rm cut} \sim \mathcal{U}(t^*_{N_{\rm peak}-1}, t_{\rm dip})$, where $T_{\rm dip}$ was defined as the unique point such that

$$t = t_{\rm dip} :\Rightarrow t^*_{N_{\rm peak}-1} < t < t^*_{N_{\rm peak}}, \frac{\partial \hat{m}}{\partial t} = 0, \frac{\partial^2 \hat{m}}{\partial t^2} < 0. \tag{6.5}$$

The truncation of only a single feature, and the requirement that a forecast be made of that single feature, was for simplicity's sake, and this made it much easier to quantify forecasting successes – besides, if at least the most imminent feature in a lightcurve is able to be successfully forecast, one could extrapolate and argue that once more data is available, and observations of *that* feature have been made, successful forecasts of the next upcoming feature will be possible, etc.

6.2.3 Algorithm setup

Hardly any changes were made to EMMA, as configured for the fitting experiments in the previous chapter. The only nontrivial change made was to the encoding of the SBLM parameter q (lens mass ratio): instead of working with $q \in [10^{-5}, 10^{0}]$, EMMA made a parameter transformation and worked with $\log_{10} q \in [-5, 0]$. The reason for this change was fairly subtle – in short, though, it amounted to assigning a proportionately larger fraction of the encoded interval $[0, 1]$ to smaller values of q. For example, using the original, 'linear' encoding, only $\sim 0.1\%$ of the encoded interval would have corresponded to values $q < 10^{-3}$, whereas with the new encoding, around 40% of the encoded interval corresponded to $q < 10^{-3}$. Since q could now span many orders of magnitude, the modified encoding simply served to improve search efficiency somewhat, by ensuring a well-balanced mix of mutation step-sizes.

The overall, fully-automated approach to making forecasts was as follows. EMMA was used to generate a very large number of candidate solutions, i.e. SBLM parameters compatible with the simulated data, based on the usual maximum-likelihood (χ^2-based) fitting approach. The fittest candidate solutions were then used to generate extrapolated lightcurves that covered the full range $(t_{\rm start}, t_{\rm end})$; these extrapolated lightcurves were then examined to see which features – if any – were predicted by the fitted models in question.

Finally, the predictions were ranked in order of fittest to poorest underlying solution, and 'duplicate' predictions were removed. Any two predictions differing by no more than $\Delta T_{\rm pred}$ were regarded as being duplicates – so, for example, if $\Delta T_{\rm pred} = 0.5$ d, predictions of a peak at $t = 100$ d and $t = 100.1$ d would've been lumped together as the 'same' prediction.

Thus, it was easy to study the best $N_{\rm pred} = 1, 2, 3, \ldots$ different predictions made by EMMA. For example, if one was interested only in a single best (at least, according to the maximum-likelihood metric) prediction, one would consider only the top-ranked

prediction; if one could afford to consider 5 different predicted scenarios, say, one would examine the 5 top-ranked predictions, etc.

In order to be able to disentangle the relative contributions of parameter-space undersampling, and genuine model ambiguities – both of which could, in principle, lead to sub-par predictions – two independent fitting sequences were run, one where fitting times were restricted to 20 minutes, and another where fitting times were restricted to 1 hour.

6.3 Results

The simulations yielded a wealth of data, and if one had the time, one could analyse a nearly endless number of relationships that exist within the data (e.g. prediction accuracy or parameter extraction accuracy vs. input parameters, lightcurve truncation points, inter-peak spacing, peak size, total number of peaks, etc.). For brevity's sake, however, only a few of the most salient results extracted from the fitting/forecasting experiments will be presented here.

6.3.1 Forecasting the time of a peak

Perhaps the first and most important question one would like to answer is: how accurately can a feature in a lightcurve (assuming, in broad terms, the setup described in Sec. 6.2) be forecast? Fig. 6.1 presents data to answer this question – unfortunately, the answer is not immediately encouraging.

If one allows only for a single prediction, we see that the probability of that prediction being accurate to within $\Delta T_{\rm pred} = 0.5$ d will only be about 20%. The probability of it being accurate to within $\Delta T_{\rm pred} = 0.25$ d will be as low as 16%, while the probability of it being accurate to within $\Delta T_{\rm pred} = 2.0$ d will be around 37%. This is better than nothing, but hardly good enough to inspire any real confidence in the forecasts.

On the other hand, if one can afford to consider multiple different predictions, the chances of success do increase. For example, if one is able to consider $N_{\rm pred} = 5$ different predictions, the chances of one of them being successful will be somewhere between 33% (for $\Delta T_{\rm pred} = 0.25$ d) and 78% (for $\Delta T_{\rm pred} = 2.0$ d). This probability of success is nontrivial, and the number of required predictions is not unreasonable (refer to the discussion in Sec. 6.4.1). Of course, as one increases $N_{\rm pred}$ and/or $\Delta T_{\rm pred}$, the probability of success should eventually increase to 100%, but the usefulness of the forecast will tend to zero, as the forecast will basically cover all future times!

On this note, Fig. 6.1 illustrates an important point. Since the probability of success is actually observed[3] to increase to 100% as $N_{\rm pred}$ is increased, we may infer that EMMA did not simply 'miss' the correct solution during its fitting sequence, in which case the probability of success would not necessarily increase to 100%. Moreover, since the overall forecasting success of the 1 hour fitting sequences was virtually identical to (only marginally better than) that of the 20 minute sequences, one can conclude that the

[3]The plots are truncated at $N_{\rm pred} = 30$, although statistics were computed all the way up to $N_{\rm pred} = 100$.

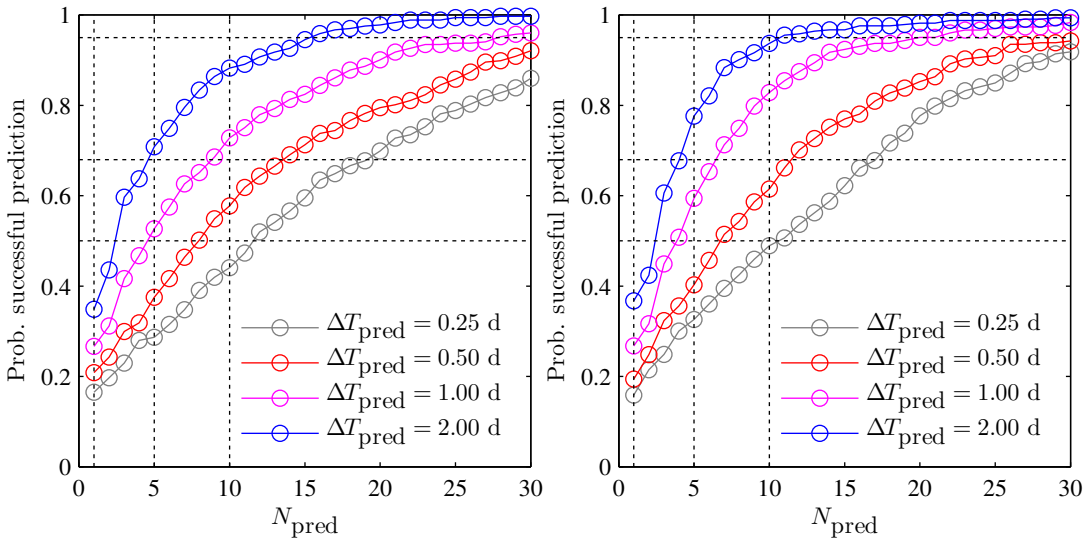

Figure 6.1 – Forecasting success vs. total number of predictions allowed, for different prediction error tolerances. The left plot is for 20 minute fitting sequences, and the right plot, for 1 hour fitting sequences. The dotted horizontal lines indicate 50%, 68%, and 95% probability thresholds. It is clear that increasing fitting times from 20 minutes to 1 hour leads to only marginal improvements in forecasting success.

relatively low forecasting success rates were due to the existence of genuinely deceptive solutions, rather than EMMA simply undersampling the relevant regions of the space. In other words, EMMA might have had no trouble locating the global optimum in a given search space: more often than not, however, the 'correct' solution would *not* have corresponded to the global optimum of that space (it might not even have corresponded to a local optimum), a problem for which the presence of missing data and noise are to blame.

Examples of successful forecasts made by EMMA are presented in Figs. 6.2 and 6.3, whilst examples of forecasts that were classified as unsuccessful are presented in Figs. 6.5 and 6.5.

6.3.2 Forecasting the existence of a peak

Given the apparent difficulty of accurately forecasting the *time* of a peak, one might ask an easier question: how successfully can one predict the mere *existence* of an imminent peak in a lightcurve, regardless of how accurately one can predict the *time* of the peak?

There are a number of conceivable ways in which the data returned by EMMA could be used to try to answer this question; perhaps the two most straightforward approaches would be:

(i) to consider whether the extrapolated lightcurve of the *fittest* solution predicts a peak; or

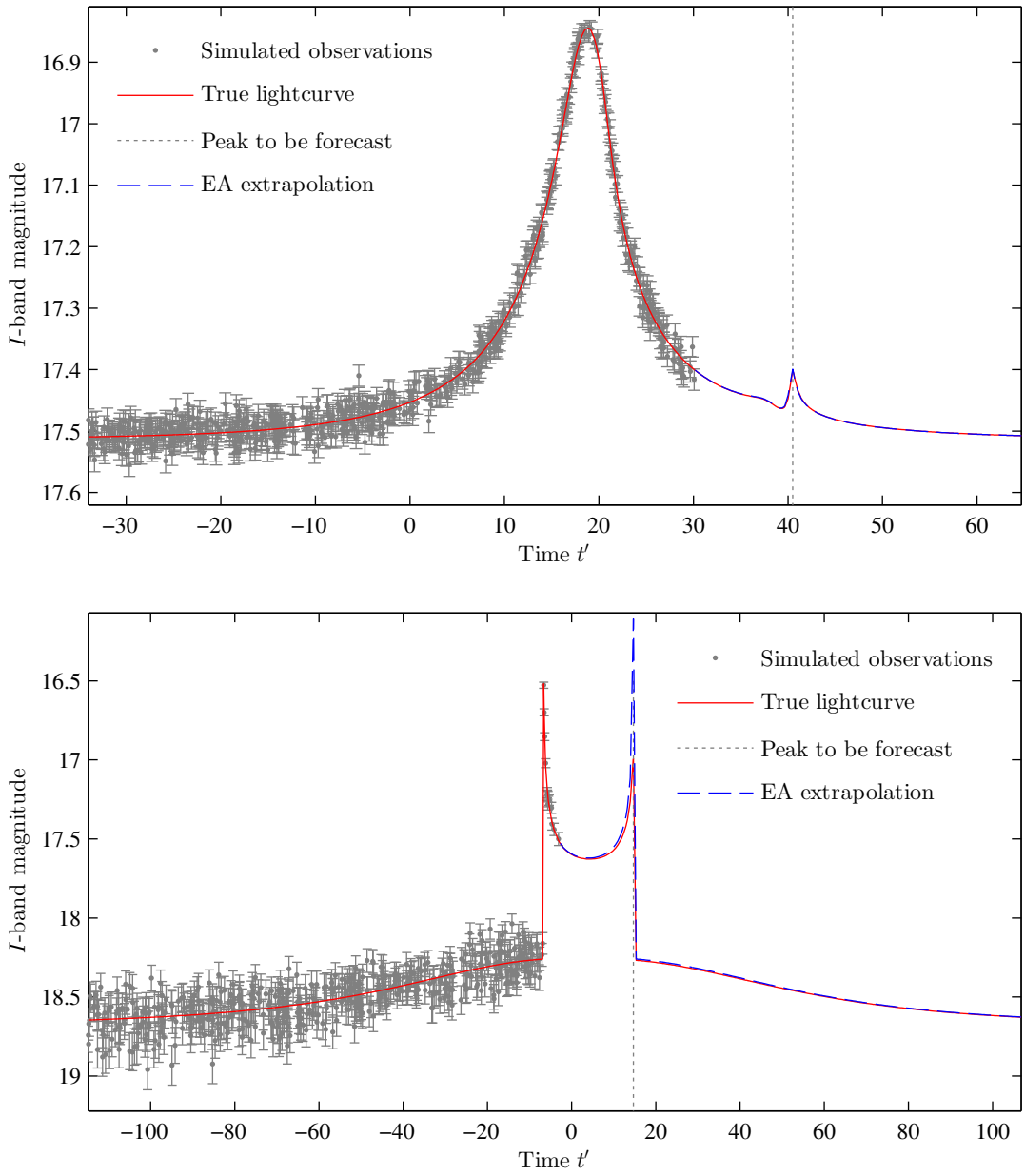

Figure 6.2 – Illustration of two forecasts that were classified as successful. In both cases, the existence of an upcoming peak, the time of the peak (to within $\Delta T_{\mathrm{pred}} = 0.5$ d), and even the morphology of the peak were all accurately predicted by EMMA.

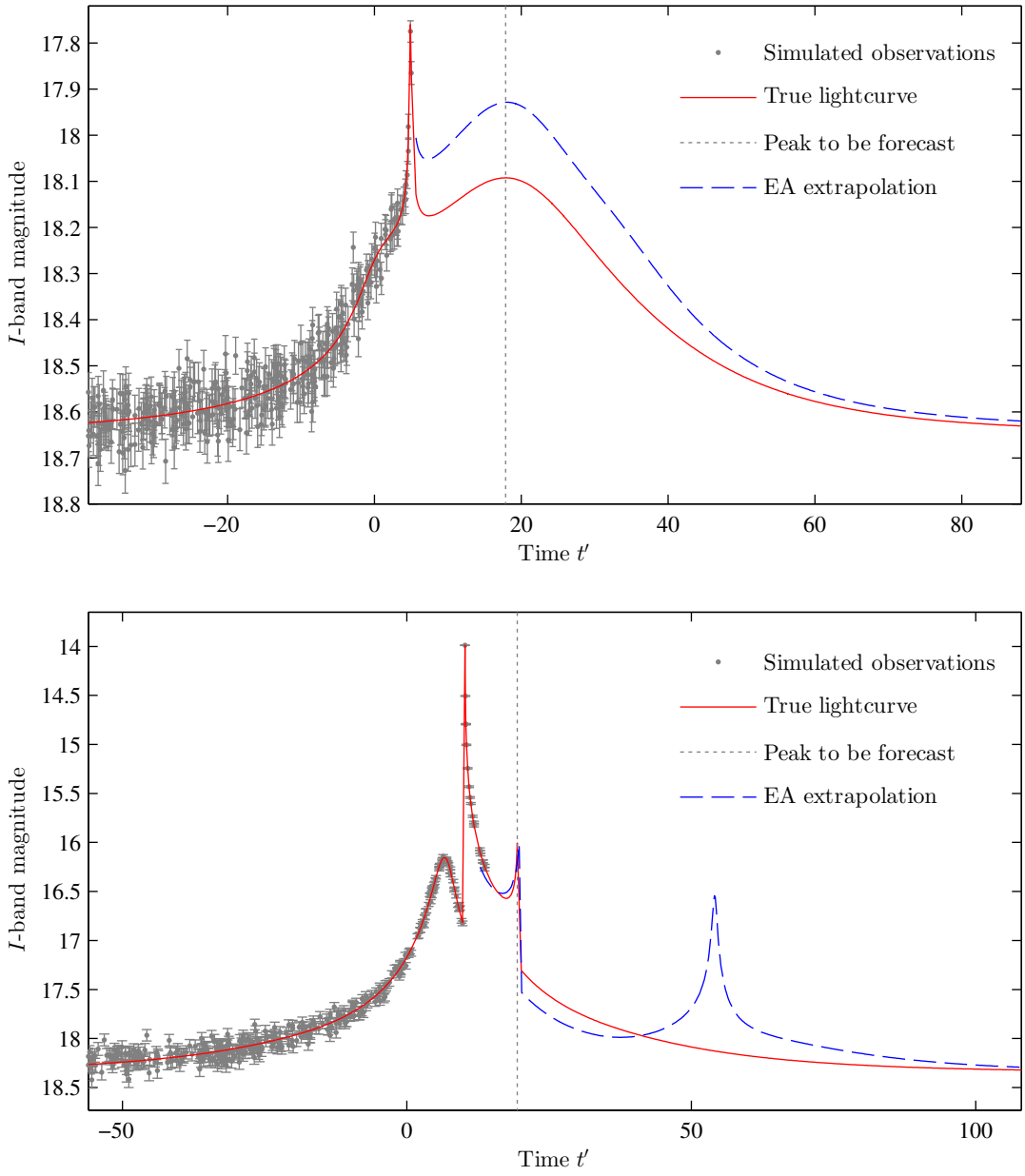

Figure 6.3 – Illustration of two further forecasts that were classified as successful. In both cases, the existence of an upcoming peak, as well as the time of the peak (to within $\Delta T_{\mathrm{pred}} = 0.5$ d) were accurately predicted by EMMA – sufficient for the forecasts to be classified as successful. However, the morphology of the predicted peak in one of the lightcurves (upper panel) differed markedly from the morphology of the actual peak; in the other lightcurve (lower panel), EMMA incorrectly predicted the existence of a second upcoming peak.

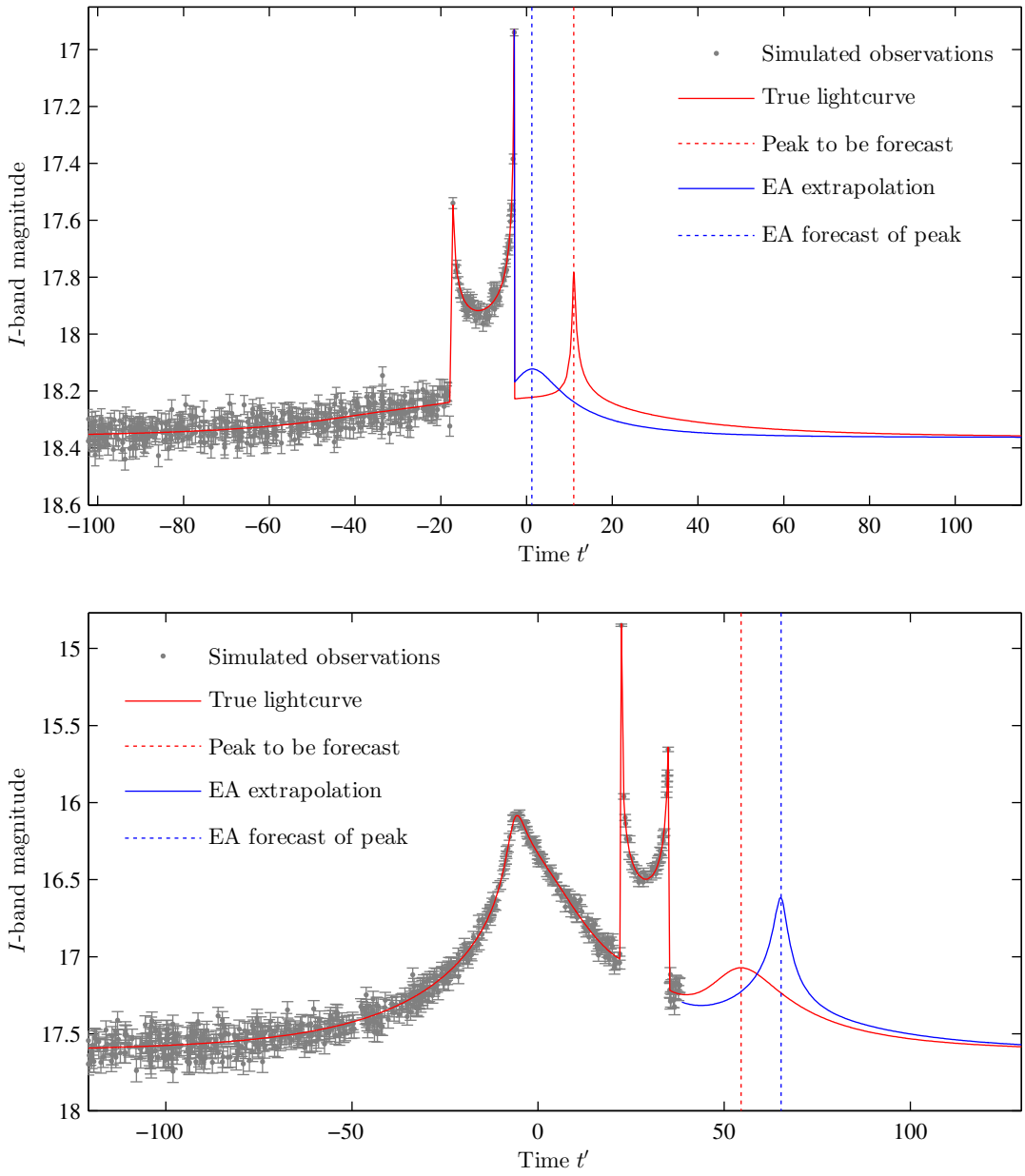

Figure 6.4 – Illustration of two forecasts that were classified as *unsuccessful*. Although the existence of a peak was correctly predicted in both cases, the predicted *time* of the peak was either too early (upper panel) or too late (lower panel) for the forecasts to be classified as successful.

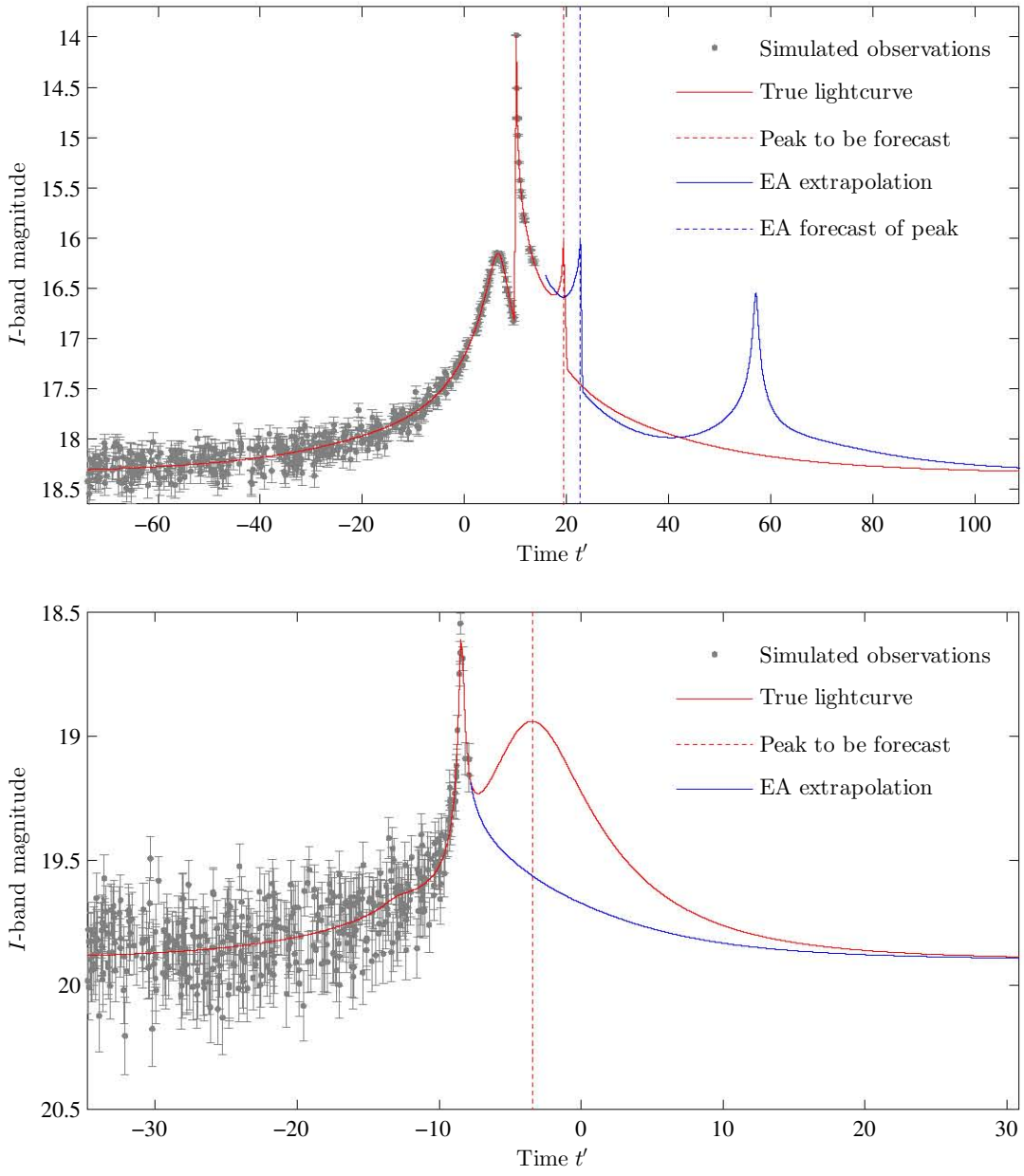

Figure 6.5 – Illustration of two further forecasts that were classified as *unsuccessful*. For the lightcurve in the upper panel, the existence of an upcoming peak was correctly predicted, but the predicted time of the peak was slightly too late for the forecast to be classified as successful; the existence of a second upcoming peak was also incorrectly predicted. For the lightcurve in the lower panel, the upcoming peak was not predicted at all.

(ii) to study the fraction of all solutions whose extrapolated lightcurves predict a peak, and if that fraction exceeds a certain threshold, to assume that a peak is, in fact, imminent.

The first approach led to a success rate of $\sim 65\%$. The second approach turned out to be slightly more fruitful, and adopting that approach, the conclusion was that one will have a $\sim 70\%$ chance of correctly predicting the existence of an imminent peak (using 80% as the threshold for the fraction of solutions that should predict a peak). While a 70% success rate is not ideal – again, noise and missing data may be blamed – it nevertheless seems promising enough so that if a peak *is* predicted, one might well think twice before 'writing-off' an event that appears to be all but completed (as may be the case if, say, a source's brightness has been in steady decline for a long time; see e.g. upper panel in Fig. 6.2). However, further tests are warranted to quantify the likelihood of making false positive predictions.[4]

6.3.3 Factors affecting the accuracy of forecasts

We return now to the original problem of forecasting the time of a peak.

The two factors which were found to have the most significant impact on forecast accuracy[5] were:

(i) the accuracy to which SBLM parameters could be extracted from a given (truncated) lightcurve; and, moreover,

(ii) the time between the truncation point and the peak to be forecast in a simulated lightcurve ($t^*_{N_{\rm peak}} - T_{\rm cut}$), i.e. the 'length' of the extrapolation required to arrive at the peak.

In order to quantify these relationships, we start by defining some new notation.

We introduce lower-case Greek deltas (δ) to denote *errors* in parameters extracted from a fit, with uppercase deltas (Δ) being reserved for quantities that denote differences between known or controllable parameters (the usage of Δs has, up to this point, anyway conformed to this unspoken convention). Accordingly, we shall use $\delta a \geq 0$, $\delta m_0 \geq 0$, etc. to denote the *relative* errors in the SBLM parameters a, m_0, and so on; we shall also use $\delta t_{\rm peak} \geq 0$ d to denote the *absolute* error in the predicted time of a given peak, assuming $N_{\rm pred} = 1$.

The correlations between the errors in various fitted parameters (δa, δb, etc.), and prediction errors ($\delta t_{\rm peak}$) are presented in Table 6.2. Note that, to ensure robust conclusions, these correlations were quantified using a standard, linear correlation coefficient,

[4] Technically, the quoted $\sim 65-70\%$ success rates refer to the probability of making a positive prediction for an event that does, in fact, contain an upcoming peak. Separate tests, using simulated lightcurves that do *not* contain any upcoming peaks, would need to be run in order to quantify the probability of making *false* positive predictions.

[5] Note: all analyses in this and the next section are based on the *combined* datasets from the 20 minute and 1 hr fitting sequences.

Parameter error	Correlation with $\delta t_{\rm peak}$			
	C_r	C_ρ	C_τ	$C_{\rm avg}$
δa	44.82%	36.95%	25.61%	35.79%
δb	38.53%	33.64%	23.30%	31.82%
δm_0	76.85%	78.13%	72.54%	75.84%
$\delta \log_{10} q$	39.90%	32.63%	22.64%	31.72%
$\delta t_{\rm E}$	43.35%	43.06%	30.35%	38.92%
$\delta t_{\rm m}$	50.79%	51.46%	37.07%	46.44%
$\delta \theta$	39.24%	33.64%	23.14%	32.00%
δf	33.59%	32.85%	31.99%	32.81%
All	45.88%	42.79%	33.33%	40.67%

Table 6.2 – Correlations between errors in fitted parameters, and errors in predicted peak times. Correlations are quantified in terms of Pearson's correlation coefficient (C_r), Spearman's rho coefficient (C_ρ), and Kendall's tau coefficient (C_τ); $C_{\rm avg}$ is an arithmetic average of the three aforesaid coefficients. The data imply that smaller errors in fitted parameters are correlated – albeit not very strongly – with more accurate forecasts.

viz. Pearson's product-moment correlation coefficient, as well as two ranked-based correlation coefficients, viz. Spearman's rho coefficient and Kendall's tau coefficient (see Press et al., 2007, for more details).

The data indicate that smaller errors in fitted parameters are generally correlated – albeit not very strongly – with more accurate forecasts. In fact, the parametric error most strongly correlated with forecast accuracy is δm_0 (correlation coefficient $\sim 75\%$), and fortunately, m_0 is also the parameter that is the easiest to fit accurately, even when dealing with incomplete lightcurves (see Sec. 2.3.4). The correlations for the other parametric errors are all positive though relatively weak (typical correlations $\sim 35\%$); this indicates that even if one manages accurately to extract most SBLM parameters from a lightcurve, an accurate forecast will *not* be guaranteed. Conversely – and somewhat more reassuringly, perhaps – the data illustrate that accurate forecasts are possible even when most SBLM parameters have not been accurately determined.

It is interesting also to study the correlations between the actual values of input lightcurve parameters, and errors in predicted peak times. Table 6.3 presents the relevant data. Surprisingly, we see that the actual SBLM parameters values that underlie a given lightcurve generally have relatively little impact on the prospects for making successful prediction. For example, the data suggest that prediction prospects will probably only be marginally better when dealing with an unblended source, and a closely-separated, stellar binary lens, than when dealing with a blended source, and star-planet lensing system. In particular, the positive but not strong correlation between m_0 and $\delta t_{\rm peak}$ indicates that photometric noise cannot be held *solely* (or even primarily) accountable for inaccurate forecasts. (Recall that, as per Eqn. 5.5, photometric errors are assumed to be larger for fainter sources.)

However, if lightcurve parameters – and the accuracies to which they are estimated –

Parameter	Correlation with $\delta t_{\rm peak}$			
	C_r	C_ρ	C_τ	$C_{\rm avg}$
a	8.10%	5.72%	4.29%	6.04%
b	12.90%	15.79%	10.80%	13.16%
m_0	26.48%	12.36%	7.85%	15.57%
$\log_{10} q$	9.46%	12.53%	8.32%	10.10%
$t_{\rm E}$	21.01%	18.62%	12.48%	17.37%
$t_{\rm m}$	6.67%	6.76%	4.27%	5.90%
θ	-1.26%	0.94%	0.44%	0.04%
f	13.20%	10.59%	6.38%	10.06%
All	12.07%	10.42%	6.85%	9.78%

Table 6.3 – Correlations between input lightcurve parameter values, and errors in predicted peak times. Correlations are quantified as in Table 6.2. The data imply that the actual SBLM parameters values that underlie a given lightcurve will have relatively little impact on the possibility of successfully predicting the future behaviour of the lightcurve.

do not have a significant impact on the probability of making an accurate forecast, what does? The answer emerges when, as suggested at the start of this subsection, we consider the relationship between prediction errors, and the lengths of time between lightcurve truncations and the peaks to be forecast (in other words, the amount of relevant data missing from a lightcurve). Accordingly, the correlation between $(t^*_{N_{\rm peak}} - T_{\rm cut})$ and $\delta t_{\rm peak}$ is illustrated in Fig. 6.6.

We see that the closer one observes to the point where a lightcurve starts to turn upwards (i.e. the point where a source starts brightening), the better one's chances of accurately forecasting the time of the associated peak. Conversely, the more data that is missing, the lower one's chances of making accurate predictions.

Thus, a coherent picture emerges. We see that forecasting success is hindered, primarily, by the fact that the lightcurves used to make forecasts are (necessarily) incomplete. In terms of the optimisation problem associated with making forecasts, the trouble stems from the fact that incomplete lightcurves place far weaker constraints on parameters than their complete counterparts; moreover, a set of SBLM parameters that is optimal (e.g. in terms of a maximum-likelihood metric) for a *complete* lightcurve will, most likely, *not* be optimal for a truncated version of that lightcurve, thus making it very difficult, if not impossible, to know which set of parameters will eventually turn out to be most compatible with the full lightcurve.

The presence of photometric noise has a similar effect (an optimal fit to a noise-free lightcurve will not necessarily be optimal for a noisy version of that lightcurve) but, assuming the best-fitting model in Eqn. 5.5, it seems to play a smaller role in hindering accurate forecasts.

Finally, noise and missing data notwithstanding, it should be remembered that binary model search spaces are inherently fraught with degeneracies and ambiguities (Sec. 2.3.4): the sets of solutions compatible even with complete, noise-free lightcurves will generally

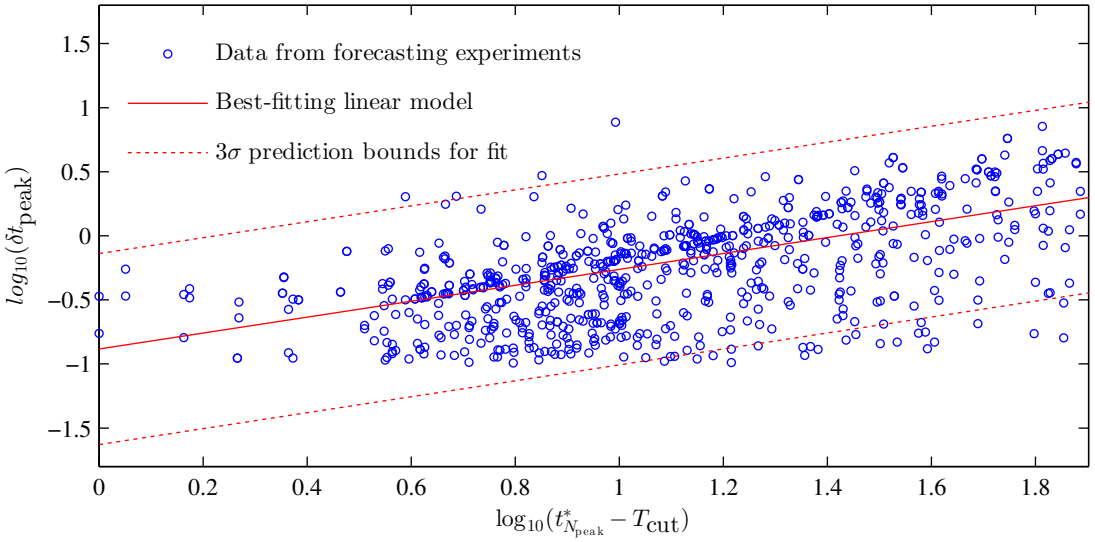

Figure 6.6 – An illustration of the strong correlation between the 'lengths' of the extrapolations ($t^*_{N_{\rm peak}} - T_{\rm cut}$) required to arrive at peaks to be forecast – i.e., the amount of data missing in the incomplete lightcurves – and the errors in the associated forecasts ($\delta t_{\rm peak}$). Correlation coefficients (C_r, C_ρ, C_τ) are all $\sim 60 - 70\%$. The data imply that the closer one observes to the point where a lightcurve actually starts to turn upwards, the better one's chances of successfully forecasting the time of the upcoming peak.

not be singletons, and therefore 100%-accurate, 'first-time-round' forecasts cannot be guaranteed, even under ideal conditions.

6.3.4 Accuracy of estimated SBLM parameters

The experiments discussed in this chapter were designed to quantify how accurately extrapolations could be made from the lightcurves of ongoing binary microlensing events, *regardless* of how closely the SBLM parameters used to generate the extrapolations corresponded to the 'correct' parameters for the events. With real events, an accurate prediction of the existence and the time of an upcoming caustic crossing, for example, would be of significant value, regardless of whether the binary parameters for the event had, in the process of generating that prediction, been very poorly estimated or constrained. Indeed, if obtaining accurate model parameters (e.g. lens mass ratios, binary separations, etc.) is one's primary concern, performing fits to incomplete lightcurves would be ill-advised, since one's prospects for accurately modelling an event will generally only be maximised when observational coverage of the event's lightcurve is maximised (Gaudi & Gould, 1997).

Nevertheless – if only as a matter of academic curiosity – one might well ask how accurately SBLM parameters can be extracted from incomplete lightcurves. Answering this question is not straightforward, and in trying to do so one could well devote

Parameter	Error distribution		
	-2σ	μ	$+2\sigma$
a	0.04%	13.06%	80.53%
b	0.03%	32.23%	165.30%
m_0	0.00%	0.28%	5.01%
q	0.09%	48.59%	898.47%
t_{E}	0.04%	11.05%	80.03%
t_{m}	0.06%	31.05%	701%
θ	0.02%	14.30%	453.94%
f	0.01%	9.48%	57.83%

Table 6.4 – Summary statistics to quantify how accurately SBLM parameters could be extracted from 1000 incomplete SBLM lightcurves (as per the setup in Sec. 6.2). The columns labelled $\pm 2\sigma$ refer, respectively, to the 2.28th and 97.72nd percentiles of the distributions of the *relative* errors in the fitted parameters; μ refers to the 50th percentiles (median values) of the distributions. Errors were compiled based on the single best solutions returned by individual fitting sequences. We may infer from the data that, for example, approx. 95% of the errors in f lie in the range 0.01%–57.83%.

an entire chapter to studying the relationships between, e.g., errors in different fitted parameters and the amount of data missing from lightcurves, the correlations between errors in different fitted parameters (e.g. how do errors in m_0 affect errors in other fitted parameters?), the amount of information available in the incomplete lightcurves (however quantified, e.g. using peak counts, or perhaps Fisher matrices – see Hundertmark, 2012), etc. For brevity's sake, then, we present in Table 6.4 only a few summary statistics to give a rough idea of how accurately parameters may be extracted from incomplete lightcurves – assuming, of course, the setup in Sec. 6.2.

From the data in Table 6.4, we see that any given parameter may, about half the time, be constrained to a relative error of $< 50\%$, and that some parameters (m_0, f, t_{E}) seem to be significantly easier to fit accurately than others (a, b, q, θ).

Only 15% of the fitting sequences returned solutions that satisfied the criterion of '$\Delta\chi^2/\nu < 1$, with errors in all fitted parameters of less than 10 percent' – this does not compare favourably with fits performed to complete lightcurves (see results in Chapter 5), where $\sim$ 95% of EMMA's fitting sequences returned solutions that satisfied this criterion.

6.4 Discussion

6.4.1 On the practicality of forecasts

The results presented in the previous section painted a fairly equivocal picture: on the one hand, it was shown that the existence of an upcoming peak in a lightcurve can be forecast quite reliably ($\sim$ 70% success rate), but on the other hand, it was demonstrated

that it is quite difficult to pinpoint the exact time of a peak.[6]

It was, however, noted that one's chances of pinpointing the time of a peak will increase as the number of distinct predictions allowed for is increased (see Fig. 6.1): this conclusion might seem self-evident, but it is not without practical value. For example, suppose one is prepared to consider $N_{\mathrm{pred}} = 5$ different predictions. Superficially, this might seem an unreasonable number of scenarios to have to consider, especially for just one microlensing event. However, it may well be the case that these 5 different predictions will be quite widely-spaced – spread out over the course of a month, say – in which case, planning extra observational coverage for the brief windows suggested by the predictions will not be unreasonable. On the other hand, predictions may turn out to be very closely-spaced, spanning just a two or three days, say; in such cases, intense observational coverage of the event *could* still be justified, provided one were reasonably confident in the predictions, and provided also that one's telescopes were not already engaged in observing very active or anomalous events.

Of course, even if one is able to ensure good observational coverage during the windows advocated by a forecast, the data in Fig. 6.1 suggest that a peak could still easily be missed – it needn't even exist in the first place (although as already noted, it isn't too difficult to make successful predictions about the *existence* of peaks).

Furthermore, as time goes by and one obtains more and more observational data, it should generally be possible to refine one's predictions, with the chances of the single best prediction being correct increasing (Fig. 6.6). For example, one might initially forecast the existence of *a* peak, yet be confronted with 5 very different, though initially equally plausible, possible times for that peak. A week later, with more observational data available for regression, the predictions may well have moved closer and closer together, some of the spurious predictions may have fallen away, etc., thus rendering the overall forecast more useful and practicable.

6.4.2 Improving forecasting accuracy

There are a number of ways in which forecasting success could, in principle, be improved.

Firstly, one needs to address the way in which one defines a successful prediction. In the experiments in this chapter, a successful prediction was defined to be one that predicted the existence of a specific 'bump' in a lightcurve, and moreover, predicted the location, to within some small error tolerance, of the *centroid* (roughly speaking) of that putative bump. This was a very specific and quite stringent criterion for success; it may be the case that a useful prediction is simply one that can, say, predict the ballpark of the time (if ever) that a 'quiescent' lightcurve will start turning upwards. It seems fair to expect better success rates when using less stringent such success criteria.

Next, one could address the actual mechanisms used to fit lightcurves, and thereby to

[6] This is consistent with the general findings of Albrow et al. (1999a) and Jaroszyński & Mao (2001); the former authors, for example, suggested that even excellent coverage of one caustic crossing is not sufficient to predict reliably the time of a second caustic crossing. It is worth noting, however, that the findings in both of the aforementioned papers were based on a very small sample of real lightcurves (one and two lightcurves, respectively).

make predictions. Throughout the fitting experiments in this chapter, predictions were informed by maximum likelihood (ML) fits, although it would be quite straightforward to use EMMA to obtain, instead, posterior-mode (maximum *a posteriori*, hereafter 'MAP') estimates of model parameters, say. In fact, Dominik (2008) showed that fits based on an MAP metric, rather than an ML metric, tend to lead to better predictions of the morphologies of single-lens lightcurves. It would be fair to expect the same to apply to binary-lens lightcurves: ML parameter estimates are, after all, well-known to be inherently biased (Press et al., 2007).

Still on the topic of mechanisms used to fit lightcurves, there are a number of data-mining techniques (see Vermaak, 2007, and references cited therein) that could, in principle, be used to extract gross features from lightcurves, in order to inform predictions; such an approach to making predictions would probably be more closely-aligned with black-box, ANN-based pattern recognition, say, than regression via iterative search/optimisation.

More sophisticated approaches could also be adopted to probe and to quantify the probability of any given prediction being successful. In the approach adopted in this work, predictions were simply ranked according to their underlying fitness – ranking aside, however, no attempt was made to quantify the relative likelihood of (i.e. to assign probability 'weights' to) any given prediction. If one could do so – using Bayesian model selection techniques (Kains, 2012), or perhaps by developing a model that maps estimated errors in fitted parameters to errors in predictions, or a model that correlates goodness-of-fit with reliability of predictions – it might be possible to reduce a large number of predictions to just one or two plausible predictions. It might also be possible to aggregate the information contained in different solutions to form a single, more 'definitive' prediction.

Finally, and perhaps most significantly, the fitting approach used in this chapter, as in the previous chapter, incorporated absolutely no information about the specific structure of the microlensing modelling problem, e.g. knowledge of lightcurve morphologies, or of the topologies of underlying critical curves and source trajectories, feature-based parameterisations, etc. As a simple example, observing a caustic entry necessarily implies a caustic exit (Gaudi, 2010). Incorporating such prior knowledge could definitely be used to reduce the 'full' model parameter space to smaller spaces of feasible parameters, thus reducing the number of solutions compatible with the given data (Kains et al., 2009), and ultimately eliminating a number of would-be predictions.

6.4.3 Extending this work

While real-time modelling efforts are becoming increasingly commonplace in modern observing campaigns, there remain very few quantitative analyses of their efficacy: the work presented here has laid groundwork for further quantitative studies of the modelling of active/ongoing microlensing events. Such studies could be used to improve techniques that admittedly do, already, appear to be achieving an admirable degree of success (Bozza, 2012; Han, 2012).

The scope for extensions to the work presented in this chapter is virtually endless.

Nevertheless, just one or two specific extensions would be particularly important if one wished to repeat these experiments with a view to making actionable recommendations for real-world observations.

Perhaps most importantly, the SBLM used in the forecasting experiments should be extended to include finite-source effects, and possibly also other second-order effects such as parallax and orbital motion effects. Such effects are present in many real-world events (see Sec. 6.5), and so it would probably be important to take them into account when modelling active events. The usual caveats about increased computational time – possibly to be offset by e.g. GPU-parallelisation, as discussed in the previous chapter – would apply when working with these more sophisticated microlensing models. It would also be worth investigating the impact that the inclusion of these higher-order effects will have on predictions, and in particular, which higher-order effects may be neglected without detrimental effect on forecasting accuracy. For example, we have already seen that is is possible accurately to predict the *time* of a peak, even while getting its detailed morphology wrong (see e.g. Fig. 6.3), in which case it should sometimes (often?) be possible to make accurate predictions even if ignoring manifest finite-source effects[7] In fact, the next section provides a few examples to prove that this is indeed possible, although rigorous studies are still necessary to figure out exactly when various higher-order effects need to be incorporated, or when they may be neglected.

The previous subsection listed a number of ways in which the approach to making forecasts could be changed, in order to arrive at more useful and/or accurate forecasts. These suggestions could all be investigated in future work. Moreover, even the data generated in the forecasting experiments of this chapter could be further mined to try to find more potentially informative relationships: for example, the relationships between temporal sampling density and prediction accuracy, between goodness-of-fit and prediction accuracy, and so on, could be investigated.

Finally, if one wanted a forecasting mechanism to form a part of a truly automated system, it would probably be necessary to implement forecasting mechanisms for dealing with single-, binary-, and even triple-lens events. Of course, the basic algorithm used to fit lightcurves needn't differ for different types of events, provided it were equipped with the functionality to access different possible models – indeed, it should take mere minutes to equip EMMA to fit triple-lens lightcurves in addition to binary-lens lightcurves, say, although the actual modelling of triple-lens lightcurves will likely be a very slow process. Given an arbitrary microlensing lightcurve, then, the *lex parsimoniae* (a.k.a. *Occam's razor*) would dictate that an autonomous fitting-system should first try to perform a fit using the simplest possible model, viz. a single-lens model. If an adequate fit to the data can be obtained, great; if not, the next-most complex model, viz. the binary-lens model, should be tested; and so on. It would not be very difficult to implement such a system computationally.

[7] Recall that, more often than not, the finite source effect only will only affect the morphology of caustic-crossing regions of a lightcurve, corresponding to localised, high amplification peaks.

6.5 A few real-world fits

The previous chapter made impressive claims, based on fitting experiments using simulated lightcurves, about EMMA's speed and robustness, and the bulk of the present chapter has been devoted to using EMMA to fit simulates of incomplete lightcurves. Simulations are all very well, but the key question is: will the algorithm's proverbial wheels come off when it is confronted with real data, and its myriad subtle complications (systematic errors, inexplicable outliers, unusual temporal samplings, higher-order microlensing effects, etc.) not captured by simple simulations?

This dissertation's *pièce de résistance* – or at any rate, the culmination of the preceding chapters – then, is a demonstration of EMMA's ability not just to perform fits to, but also to generate predictions from, *real* binary-lens lightcurves. For brevity's sake, and on account of limited space in this dissertation, only a couple of representative examples of EA-based forecasts are presented here.[8]

6.5.1 OGLE-2000-BUL-46

The first event we consider is OGLE-2000-BUL-46, a microlensing event discovered in 2000 by the OGLE-II group (Udalski et al., 1997). This event was chosen because it has already been the subject of a quantitative forecasting experiment (Jaroszyński & Mao, 2001), and also because of its clear binarity, and relatively low photometric noise.[9]

To prepare the lightcurve, all data outside $-60.0 \leq t' := \mathrm{HJD} - 2451750 \leq 40.0$ d were discarded; this amounted to discarding the quiescent, zero-magnification parts of the lightcurve (the low-magnification tails serve to constrain m_0, but otherwise have no real impact on binary parameters). Thus, about 100 days' worth of data, more or less centred on the 'interesting' parts of the lightcurve, remained. Fig. 6.7 refers.

The full lightcurve contained three distinct peaks: one at $t' \sim -12.0$ d, a second at $t' \sim -5.1$ d, and a final, well-sampled one at $t' = 2.9$ d. Following the procedures described in Sec. 6.2, the lightcurve was truncated at $t' = -3.0$ d, i.e. approximately 6.0 d before the final peak, and about 4.0 d before it would have been apparent that the lightcurve would return to an upward trajectory. Finally, then, the data spanning $-60.0 \leq t' \leq -3.0$ d were passed to EMMA, whose goal it was to try to predict the existence and the time of the peak at $t' = 2.9$ d.

As usual, fitting was based on a χ^2-metric, and the parameter ranges in Table 6.1 were assumed. Because the total number of datapoints to be fitted was quite small (~ 100), EMMA's fitting sequence was restricted to 20 minutes – in fact, EMMA managed to locate solutions compatible with the data in far less time, viz. on the order of two minutes.

As in Sec. 6.2, all solutions returned by EMMA were automatically binned into groups of distinct predictions (i.e. predicted peak times), each group defined by its fittest mem-

[8] Moreover, as is noted in Sec. 6.6, a larger sample of lightcurve forecasts will not, by itself, allow one to quantify rigorously EMMA's utility in generating predictions from real-world lightcurves.

[9] Original photometric data for the event is available online at `ftp://ftp.astrouw.edu.pl/ogle/ogle2/ews/2000/bul-46/`.

ber, and with the prediction of the defining member having to be separated from the predictions of the defining members of other groups by at least $\Delta T_{\rm pred} = 0.5$ d. With this done, almost all solutions were found to fall into one of the three fittest groups; in descending order of the fitness of their defining solutions, these groups were denoted Forecasts 1, 2, and 3. The best solution found by EMMA, i.e. the solution or lightcurve defining Forecast 1, satisfied $\chi^2 \sim 1.24$ per degree of freedom, for the incomplete lightcurve. For comparison, Jaroszyński & Mao obtained a fit satisfying $\chi^2 \sim 1.11$ per degree of freedom, when using an extended-source model to fit the *complete* lightcurve.

It turned out that the solutions in all three groups predicted the existence of a peak: in order, Forecasts 1, 2, and 3 predicted peaks at $t' = -0.6$ d, at $t' = 0.9$ d, and at $t' = 3.0$ d. Success! The *third* forecast turned out to be the 'correct' one, predicting the time of the real peak to within 0.1 d. The lightcurves associated with each of the three groups are illustrated in Fig. 6.7. Interestingly, Jaroszyński & Mao claimed that it was only possible reliably to constrain (to within ± 0.1 d) the time of the second peak ~ 1 d before the actual caustic crossing – crucially, though, their work did not allow for the possibility of multiple, ranked predictions.

The fitted models underlying all three of EMMA's forecasts favoured a moderately-blended ($f \sim 65 - 75\%$) source, lensing an intermediate-separation binary ($a \sim 1.5$). There appeared to be a degeneracy between the parameters b and q, with both a close source-approach and a high-mass secondary ($b \sim 0.1$, $q \sim 0.9$), and an intermediate source-approach and an intermediate-mass secondary ($b \sim 0.5$, $q \sim 0.4$), being compatible with the data. In the fitted models, the three peaks corresponded, in order, to a cusp approach, a caustic entry, and a caustic exit.

The example of OGLE-2000-BUL-46 serves to illustrate a number of important points raised earlier in the chapter. Firstly, we see the value of considering more than one prediction: if we had considered just the single 'best' prediction, we would have been correctly informed about the existence of an imminent peak (corresponding to a caustic crossing, as it turned out), but the predicted time of the peak would have been off by about 3 days. Following up on just two further predictions, however, would have enabled the time of the peak to be pinpointed to within ~ 2 h, suggesting a favourable cost-benefit ratio for the extra predictions.

Secondly, we see that the presence of missing data and noise can render *sub-optimal* a solution that would have been optimal for the complete lightcurve. Referring to Fig. 6.7, it is clear that solution 3 is more compatible with the full dataset than solutions 1 and 2, and yet it is not the solution most compatible with the truncated lightcurve.

Finally, it is interesting to note that a successful forecast was possible even though the underlying modelling completely ignored the (possibly quite weak) finite-source effects present in the data. Indeed, it is clear that the SBLM does not facilitate a perfect fit to the data – a good fit is achieved to the first peak, which corresponds to a cusp approach, but the detailed morphologies of the second two peaks, both of which correspond to caustic crossings, are not reproduced very accurately.

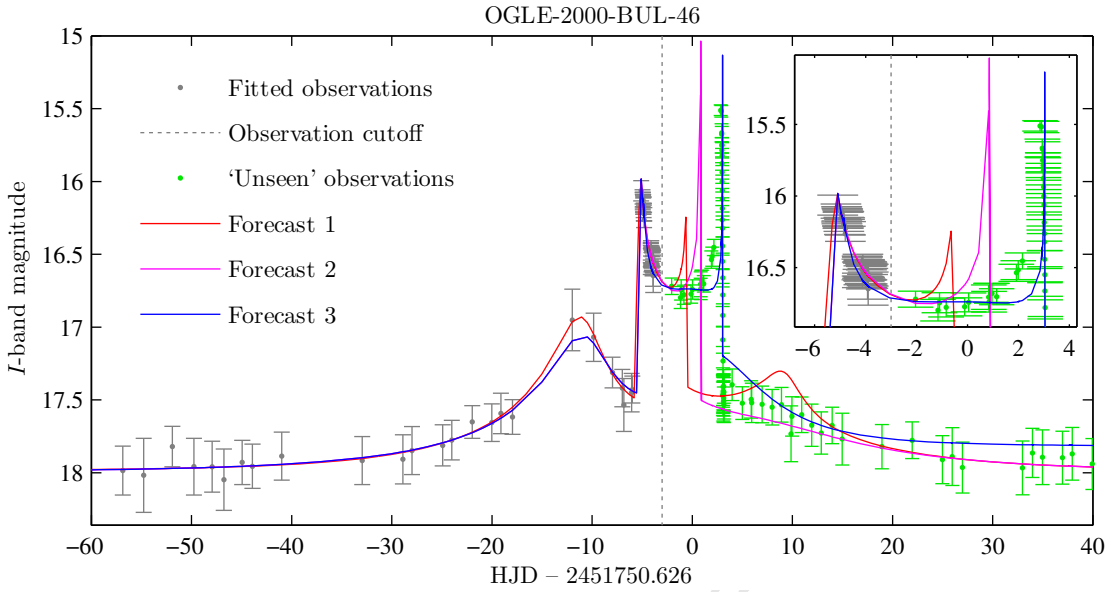

Figure 6.7 – Lightcurve for OGLE-2000-BUL-46. The gray and green points represent (a subset of) the original observational data, plotted along with 3σ error estimates. Fits were performed to the gray datapoints only; all three resultant forecasts correctly predicted an imminent caustic crossing, although only the third forecast accurately ($\delta t_{\rm peak} \sim 0.1$ d) predicted the time of the crossing.

6.5.2 MOA-2011-BLG-197/OGLE-2011-BLG-0265

The next lightcurve we consider is that of MOA-2011-BLG-197, a.k.a. OGLE-2011-BLG-0265, observed in 2011 by both the MOA and OGLE collaborations. The event was chosen because the best available models for the event suggested a low- or planetary-mass secondary lens ($q \sim 10^{-3}$), and also because the event's lightcurve contained far more data than the previous one considered – the OGLE dataset alone constituted some 3500 observations, spread over a year and a half – and so performing fits to this lightcurve provided EMMA with a decidedly nontrivial computational workload.[10]

Data outside the range $-100 \leq t' := \text{HJD} - 2455760 \leq 100.0$ d were discarded; again, this amounted to excluding the zero-magnification tails of the lightcurve from the fitting. The full lightcurve contained two distinct, well-sampled peaks: one at $t' \sim -13.3$ d, and a second at $t' = 0.6$. Following the procedures described in Sec. 6.2, the lightcurve was truncated at $t' = -12.0$ d, i.e. approximately two weeks before the final peak, and about two days before it would have been apparent that the lightcurve would return to an upward trajectory. Fig. 6.8 refers.

Following a 2-hour fitting sequence, EMMA returned a handful of solutions satisfying

[10] Original photometric data for the event are available online at `ftp://ftp.astrouw.edu.pl/ogle/ogle4/ews/2011/blg-0265/`, `https://it019909.massey.ac.nz/moa/alert/display.php?id=gb10-R-3-16297`, etc.

$\chi^2 \sim 20$ per degree of freedom: nominally this was a mediocre result, although the fitted lightcurves did appear to match the observational data very well. It soon became apparent that the large χ^2 statistic was due, in fact, to a small number of high-magnification points with very small error estimates: either the error estimates for these points were too small, or, more likely, finite source effects prevented a perfect fit to these points using only the SBLM. In any event, after excluding these points from the fitting, the best solution, though basically identical to the best solution found initially, satisfied a more respectable $\chi^2 = 1.63$ per degree of freedom.

After solutions were binned, the second-fittest group of solutions predicted a peak at $t' = 0.4$ d: a successful prediction, even using $\Delta T_{\rm pred} = 0.25$ d. In fact, the eleven top-ranked groups all predicted a peak in the ballpark of $t' = 3.0 \pm 3.0$ d. The lightcurves associated with the three top-ranked groups of predictions are illustrated in Fig. 6.8.

Bozza (2012) actually used MOA-2011-BLG-197 as an example of an event for which successful, real-time predictions were facilitated by his 'RTModel' software. It certainly seems remarkable, however, that EMMA – a robust, general-purpose optimisation tool, not at all specialised for microlensing modelling – could replicate the success of the highly-specialised RTModel system. Even more remarkably, the basic parameters found by EMMA $(a, b, m_0, q, t_{\rm E}, t_{\rm m}, \theta, f)$ were all marginally consistent with those found by Sumi (2011), even though he used a rather more sophisticated model – including finite-source and parallax effects – to extract these parameters from the *complete* lightcurve.

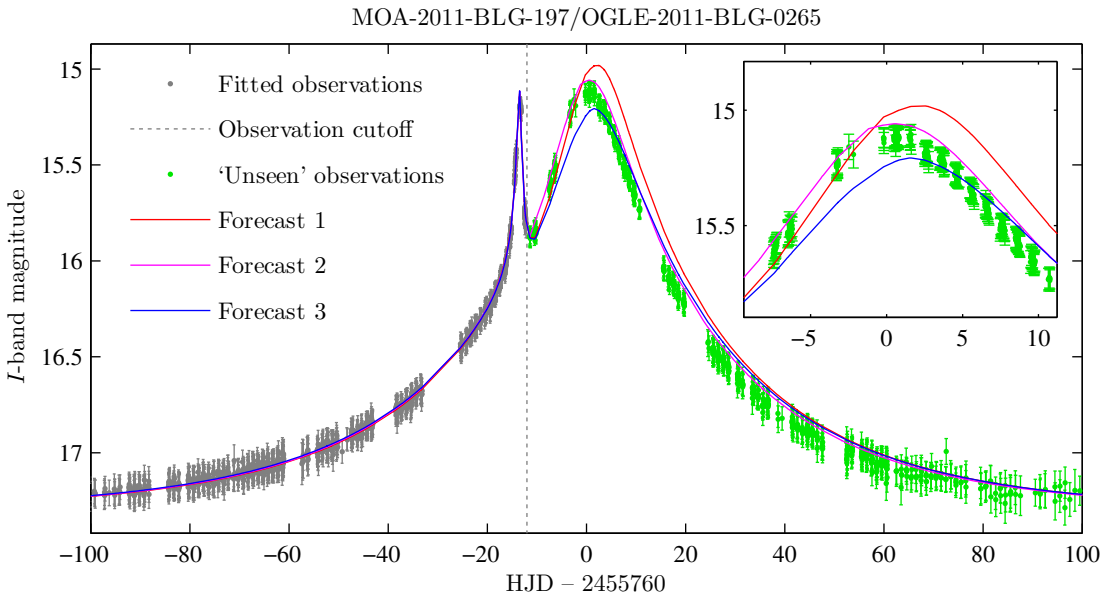

Figure 6.8 – Lightcurve for MOA-2011-BLG-197/OGLE-2011-BLG-0265. The gray and green points represent (a subset of) the original observational data, plotted along with 3σ error estimates. Fits were performed to the gray datapoints only. The fittest three forecasts are illustrated; the second forecast accurately ($\delta t_{\rm peak} \sim 0.2$ d) predicted the time of the second cusp approach.

For comparison, his models satisfied $\chi^2 \sim 1.15$ per degree of freedom. The best models favoured an unblended source ($f = 1.00$), lensed by an intermediate-separation binary ($a = 1.04$) with a low-mass secondary lens ($q = 3.3 \times 10^{-3}$); in these fitted models, both peaks corresponded to cusp approaches.

Perhaps MOA-2011-BLG-197 was an unusually 'easy' event, where even a limited, truncated lightcurve was sufficient to place fairly strong constraints on model parameters. It probably also worked in the favour of SBLM-based fitting that the event featured cusp approaches rather than caustic crossings, meaning that finite-source effects were not too pronounced.

Finally, it was interesting (and initially, more than a little puzzling) to note that EMMA was actually able to extract parameters far more consistently and speedily from the *incomplete* lightcurve than the complete one. The explanation turned out to be quite straightforward: for the complete lightcurve, relatively good solutions could be found that captured only the second, very broad peak at $t' = 0.6$ d, with little penalty (in terms of χ^2) for missing the narrow peak at $t' = -13.3$ d. With the incomplete lightcurve, on the other hand, solutions that did not reproduce the $t' = -13.3$ d peak were critically penalised – and fortunately, solutions that captured the $t' = -13.3$ d peak generally also seemed to capture the latter, broad peak.[11] This indicates that it might be worth investigating the more general possibility of manually down-sampling specific features in – if not outright excluding from – the lightcurves used to perform fits, in order to enhance fitting speed (see also Hundertmark, 2012).

6.5.3 OGLE-2011-BLG-0417

The final event was chosen – in part, at least – to serve as an example of an event where EMMA's forecasting success was only marginal, and thus to offset the unqualified successes suggested by the previous two examples. The event in question is OGLE-2011-BLG-0417, discovered in 2011 by the OGLE-IV Early Warning System (Udalski, 2003).[12]

In-depth modelling of the event was carried out by Shin et al. (2012); second-order effects, including parallax and lens orbital motion, were included in the modelling effort. In fact, the 'standard' binary model used by Shin et al. – basically the SBLM, with provision made for finite source effects – yielded $\chi^2/\nu = 4415/2627$, or $\chi^2/\nu = 1.68$; the incorporation of the parallax effect led to $\chi^2/\nu = 0.91$, and the further addition of orbital motion effects led to $\chi^2/\nu = 0.66$, suggesting a superb fit.[13] The conclusion, then, was

[11] Adopting a set-theoretic view, we may denote the set of trial solutions that captures the first peak $\mathcal{S}_1$, and the set of trial solutions that captures the second peak, $\mathcal{S}_2$. Then, apparently $\mathcal{S}_1 \subsetneq \mathcal{S}_2$, so that $X \in \mathcal{S}_1 \rightarrow X \in \mathcal{S}_2$; hence, performing fits exclusively to the first peak seems to guarantee that the second peak will also be captured.

[12] Original photometric data for the event are available online at `ftp://ftp.astrouw.edu.pl/ogle/ogle4/ews/2011/blg-0417/`.

[13] For large ν, $\chi^2 \sim \mathcal{N}\left(\nu, \sqrt{2\nu}\right)$ – hence the popular rule-of-thumb that $\chi^2 \sim \nu$, or $\chi^2/\nu \sim 1$, is indicative of a good fit. The $\chi^2/\nu = 0.66$ obtained by Shin et al. (2012) is almost surely indicative of overestimated, and/or non-Gaussian, photometric errors (Press et al., 2007), since, with $\nu \sim 2600$, $\chi^2/\nu = 0.66$ represents a 12σ deviation from the expected value!

that the event's lightcurve was significantly affected by both parallax and orbital motion effects. Nevertheless, the lightcurve was passed on to EMMA, which bravely (naïvely!) attempted SBLM-based fitting.

Initial preparation of the lightcurve entailed discarding data outside $-60 \leq t' := \text{HJD} - 2455815 \leq 60.0$ d. Although the lightcurve of the best-fitting models found by Shin et al. (2012) contained three peaks, only the first of these peaks was fairly well-sampled by the OGLE data, with only the leading edge (i.e. rise) of the putative second peak, and the trailing edge (i.e. fall) of the putative third peak, being sampled. The observational data do clearly indicate that the first peak occurs at $t' \sim -15.0$ d, while the models of Shin et al. suggest that the latter two peaks occur at $t' \sim 5.0$ d, and $t' \sim 7.1$ *d*. A mere visual inspection of the OGLE data, however, would suggest – misleadingly, apparently – a broad peak at $t' \sim -15.0$ d, and a narrow peak somewhere around $t' \sim 6.0$ d. Fig. 6.9 refers.

In a slight variation of the procedure described in Sec. 6.2, the lightcurve was truncated at $t' = -9.5$ d: more than two weeks before the final two peaks, and nearly one week before it would have been apparent that the lightcurve would return to an upward trajectory. For the first time, then, *two* features, rather than just a single a feature, were truncated from the lightcurve; the reason for doing this was twofold. Firstly, the putative peak at $t' \sim 7.1$ d was not directly sampled by the available observational data, and so it would have been difficult to compare agreement between predictions and 'reality' (other than by assuming the various models of Shin et al.) On the other hand, since all forecasting experiments up to this point had been based on predicting single features, it was decided that it would be interesting to examine – albeit using a once-off, non-rigorous test – how EMMA would fare at predicting the existence of *two* peaks.

Initial fits performed by EMMA (after a 1 hour fitting sequence) were nominally unsuccessful, with the goodness-of-fit of the best solutions being $\chi^2 \sim 100$ per degree of freedom. Consequently, as for the previous lightcurve, the points making the largest contribution to the total χ^2 were excluded from the fitting. With this done, fits satisfying $\chi^2/\nu \sim 1.8$ were achieved.

Following the usual binning/grouping of solutions, the results were decidedly mediocre. The fittest group of solutions predicted that there would be no upcoming peak at all – let alone two. The second- and third-ranked groups of solutions predicted the existence of a single peak, but unfortunately the predicted times were nowhere near $t' \sim 6.0$ d. The fourth-ranked group did, correctly, predict two upcoming peaks, although it predicted the times of these peaks as being $t' = 1.7$ d $t' = 5.0$ d, meaning that it matched only the first of the upcoming peaks, and even then, this match seemed to be rather serendipitous (refer to Fig. 6.9). The seventh-ranked group predicted a single peak at $t' = 4.7$ d, i.e. in the ballpark of the real $t' = 5.0$ d peak, but it too failed to predict the existence of the second imminent peak. In fact, none of the ten top-ranked groups of solutions predicted a peak within $\Delta T_{\text{pred}} = 0.5$ d of $t' = 7.1$ d, and not a single solution predicted accurately the times of both peaks.

Were the date today $t' = -9.5$ d, and were one wondering what the lightcurve of OGLE-2011-BLG-0417 might do in the near future, it'd be debatable whether the predictions made by EMMA would have been of any value at all.

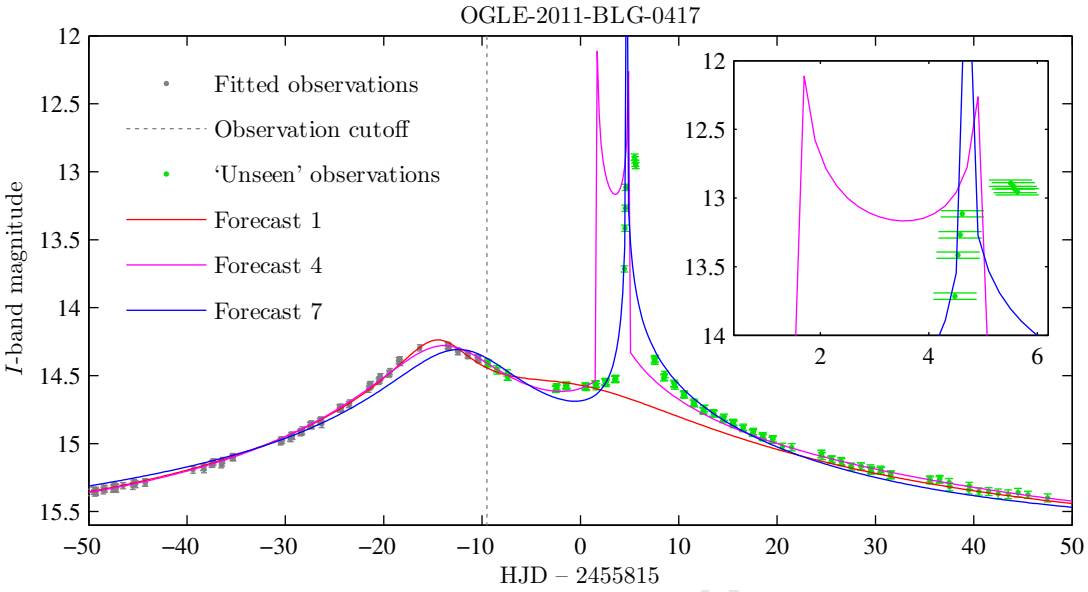

Figure 6.9 – Lightcurve for OGLE-2011-BLG-0417. The gray and green points represent (a subset of) the original observational data, plotted along with 3σ error estimates. Fits were performed to the gray datapoints only. The fourth- and seventh-ranked forecasts accurately predicted the time of the $t' = 0.5$ d peak, though unfortunately, no forecasts accurately predicted the existence of a peak $t = 7.1$ d. Note that the latter peak's existence is implied by the models of Shin et al. (2012); its presence is not obvious from a visual inspection of the observational data.

On the one hand, the poor forecasting accuracy for this event does not, by itself, warrant cause for concern. As shown in Sec. 6.3, even in ideal cases where lightcurves can be described perfectly by the SBLM, accurate forecasts cannot be guaranteed – and of course, this lightcurve featured a number of second-order (finite-source, parallax, orbital-motion) effects not accounted for by the SBLM, so conditions were far from ideal. On the other hand, however, Bozza (2012) listed OGLE-2011-BLG-0417 as an event for which the RTModel software *was* able to make an accurate prediction (it is not clear how far in advance RTModel was able to forecast relevant lightcurve structures; to be fair, EMMA was given the fairly difficult task of forecasting two peaks, more than two weeks in advance) – this suggests, rather unsurprisingly, that there are events for which forecasting accuracy will suffer on account of the approximations made by the SBLM. Moreover, since there will generally be little way of knowing in advance whether it would be acceptable to neglect second-order effects from modelling, the conclusion is that SBLM-based forecasts should probably only be used to produce 'rough', first-order fits (as may be desirable when pressed for time, e.g. when modelling short-timescale events). Of course, this conclusion is in no way an indictment of EMMA: the algorithm itself should be able to move seamlessly from SBLM-based fitting to fitting using any one of a number of more sophisticated models.

6.6 Conclusions

With more and more microlensing events being observed every year, and with these events being observed at ever higher cadences, the volumes of data that need to be analysed are growing at alarming rates. It is therefore becoming more important than ever before to develop systems that can perform accurate and autonomous modelling[14] of microlensing events, and this holds especially true for *active* microlensing events, where autonomous, real-time modelling could be used to generate forecasts that inform future observations of the events.

In recent years, such real-time modelling efforts have become increasingly commonplace, and although such modelling efforts have already started to bear fruit, there have unfortunately been very few quantitative analyses of the accuracy of their forecasts. Accordingly, this chapter presented fitting experiments that quantified, in broad terms, the efficacy of making real-time fits to incomplete binary lightcurves, in order to forecast upcoming features in the lightcurves. Concurrently, the fitting experiments were also used to provide proof of concept for the use of an evolutionary algorithm to model active microlensing events.

In particular, it was shown, using an evolutionary algorithm (EMMA) and simulated binary lightcurves, that the existence of an upcoming peak in a lightcurve can be forecast quite reliably ($\sim 70\%$ success rate), but that it is difficult to pinpoint the time of such a peak (typical success rates $\lesssim 50\%$, with the exact number varying according to the requisite precision of forecasts). It was noted, however, that one's chances of pinpointing the time of a peak will increase as the number of distinct predictions allowed for is increased (to within reasonable limits), and also as one 'fills in' a lightcurve by obtaining more observational data.

Though the scope of the aforementioned fitting experiments was quite narrow, a number of extensions to the experiments were suggested, along with possible strategies – perhaps to be investigated in future work – for improving forecasting accuracy.

Finally, following the extensive fitting experiments using simulated lightcurves, a handful of *real* binary-lens microlensing events were considered, and EMMA was vindicated by demonstrating – at last – that it is indeed possible to use the algorithm not only to perform fits to, but also to generate predictions from, the lightcurves of real microlensing events. Examples of successful, mediocre and poor predictions were presented.

In principle it would've been possible to give EMMA a much larger sample of lightcurves from which to attempt to generate predictions, and thus to quantify more broadly EMMA's utility in generating forecasts. However, in order to disentangle the shortcomings of the SBLM from any actual shortcomings of EMMA, the SBLM would've needed to have been supplanted by a more sophisticated model that allowed for finite source effects, parallax effects, etc. Unfortunately, given the enormous increase in computing times that this would've entailed, such an undertaking was beyond the scope of this dissertation.

[14] Even if humans could sift through all the data, autonomous modelling would still be preferable, in order to eliminate human bias from the modelling.

7 Conclusions

This dissertation presented a new algorithm that was developed to perform autonomous fitting of gravitational microlensing lightcurves. The new algorithm, which combines features of extant evolutionary algorithms with some novel ones, was demonstrated to fare well on the difficult problem of fitting binary-lens microlensing lightcurves. In particular, it was shown that the new algorithm is capable of outperforming both conventional (e.g. iterated simplex) and more state-of-the-art (e.g. artificial neural network-based) approaches to fitting binary lightcurves. It was also shown that the new algorithm can perform fits rapidly whilst maintaining a fairly modest computational footprint, and whilst also offering a number of other appealing features, such as conceptual simplicity, versatility, robustness, and trivial parallelisation. Finally, it was demonstrated that the algorithm is capable – at least, to the extent permitted by the inherent degeneracies and nonlinearities associated with microlensing models – of using incomplete microlensing lightcurves to forecast upcoming features, and the algorithm was used to derive some quantitative results related to the feasibility of making such forecasts in general.

7.1 Review

Chapter 1 served as an introduction, and set the stage for the rest of the dissertation. In particular, the main goal of the dissertation was affirmed: to provide proof of concept for using an evolutionary algorithm as the basis for real-time, autonomous modelling of ongoing microlensing events, with a particular view to informing future observations of such events.

Chapters 2 and 3 constituted, for the most part, reviews of literature relevant to the original work presented in the later chapters.

Chapter 2 gave a brief overview of gravitational microlensing and, in particular, of the binary-lens model assumed throughout this work. It was noted that while microlensing is an important and well-established technique for discovering exoplanets, it remains notoriously difficult to model the lightcurves of microlensing events.

Chapter 3 introduced the concept of nonlinear optimisation, and presented evolutionary algorithms (EAs) as a promising means for solving difficult nonlinear optimisation problems. The chapter discussed the ideas underpinning EAs, outlined some of their advantages and disadvantages – with particular emphasis on their suitability for microlensing modelling – and gave examples of the many applications that these algorithms have

already found in astronomy and astrophysics.

Chapter 4 introduced and discussed, in some detail, the new evolutionary algorithm, 'EMMA', developed by the author of this dissertation for the purpose of modelling microlensing events, as well as for solving more general numerical optimisation problems. The algorithm was noted to be versatile, robust, conceptually straightforward, highly customisable, and highly amenable to parallelisation.

Chapter 5 presented fitting experiments that were set up and run to illustrate the performance of EMMA, as well as to provide proof of concept for using EAs to model the lightcurves of completed microlensing events. It was argued in this key chapter that EAs do, in fact, seem well-suited to modelling binary-lens microlensing lightcurves: despite the difficulty of the fitting problem, EMMA yielded excellent fitting accuracy whilst maintaining a relatively modest computational footprint, and also offering a number of other desirable properties.

Finally, Chapter 6 extended the work of the previous chapter and presented fitting experiments (using more realistic data than in the previous experiments) that were set up and run to quantify, in broad terms, the efficacy of making real-time fits to incomplete binary lightcurves, in order to forecast upcoming features in the lightcurves. It was shown, using EMMA, that the existence of an upcoming peak in a lightcurve can be forecast quite reliably, but that – largely on account of inherent ambiguities associated with microlensing models – it is far more difficult to pinpoint the time of such a peak. It was noted, however, that one's chances of pinpointing the time of a peak will increase as the number of distinct predictions allowed for is increased (to within reasonable limits), and also as one 'fills in' a lightcurve by obtaining more observational data. Lastly, it was demonstrated that it is indeed feasible to use an EA not only to perform autonomous fits to, but also to generate useful predictions from, the lightcurves of *real*, ongoing microlensing events, including possible planetary events. Thus, the goal set out at the start of the dissertation was achieved.

7.2 Possibilities for future work

In considering possible extensions to the work presented here, one turns naturally to the simplifying assumptions made in this work. (Whether this work actually merits extension is considered in the closing remarks contained in the next section.)

Without doubt, the single most critical and lamentable simplifying assumption made in this work is that the SBLM (the simple binary-lens model introduced in Chapter 2) provides an adequate description for real binary microlensing events. To be sure, the SBLM is a nontrivial and unquestionably powerful model, and has proved quite successful in the interpretation of a number of real microlensing events, including planetary events; as demonstrated in Chapter 6, the model can often even be useful for generating forecasts based on fits to lightcurves of ongoing microlensing events. On the other hand, there are number of instances in which the SBLM is known to be an inadequate model – for example, when dealing with lightcurves where finite-source effects are pronounced – and in such instances, it becomes imperative to extend the basic model.

A good deal of rationalisation was provided throughout this dissertation for adopting the SBLM, though perhaps the primary impetus for its adoption was the limited timeframe associated with the author's masters project (of which this dissertation serves, of course, as a chronicle). For example, even with the respectable computing power available to the author, an experiment featuring fits to about a thousand simulated lightcurves took on the order of two weeks to run, and many such experiments needed to be run during the development and testing of various algorithms. If an extended-source model were to have been used for fitting, just one such experiment would've taken more than two *years* to run! Of course, it was argued in this work that high-performance computing platforms such as GPUs could be used to reduce an EA's fitting times by orders of magnitude – unfortunately, efficiently porting existing code to a GPU was found to be a nontrivial task, the successful completion of which would've required significant experience with the GPU programming paradigm.

In any event, though the SBLM may have been 'good enough' for this proof-of-concept work, the case for EA-based microlensing modelling could be made far more compelling if an EA – whether EMMA or some other algorithm – could be demonstrated to yield strong performance even when working with a more sophisticated microlensing model than the SBLM.

A number of secondary, smaller-scale extensions to the fitting experiments presented in this work were suggested in Chapters 5 and 6. These suggestions will not be rehashed here – at any rate, the suggestions would probably only be worth acting on if the SBLM were first to be supplanted in the fitting experiments by a more realistic and comprehensive microlensing model.

7.3 Final words

Though this dissertation extolled the many virtues of EAs, and though it provided proof of concept for autonomous, EA-based microlensing modelling and forecasting, some words of circumspection are in order.

As already mentioned, although the performance of the EA developed for this work is promising, it remains to be seen how the algorithm will fare when faced with more difficult microlensing modelling problems. Moreover, as of 2012, there exist already a small handful of bespoke algorithms capable of performing fairly rapid fits to the lightcurves of active binary events (e.g. Bennett, 2010; Bozza, 2012): so what, then, is the relevance of this work?

First, EMMA, the algorithm introduced in Chapter 4, is a general-purpose optimisation tool with the potential to be used to solve a broad spectrum of difficult problems. Even if the actual algorithm finds few adopters, the exposition of the algorithm, along with the general treatise on EAs given in Chapter 3, serves to promote awareness about the existence and relevance of EAs. Next, the experiments presented in Chapter 5 constitute the first (at least, to the author's knowledge) in-depth study of EA-based microlensing modelling; as already noted, the content of both Chapter 3 and Chapters 4–5 of this dissertation have been published (in modified form) in two separate, peer-reviewed

publications, viz. Rajpaul (2011) and Rajpaul (2012), respectively.

The work presented in Chapter 6, though it represents one of the few quantitative studies of regression-based microlensing *forecasting*, is not entirely novel; nevertheless, the chapter lays the groundwork for further quantitative studies of the modelling of ongoing microlensing events, and it does culminate in some actionable (albeit tentative) recommendations to microlensing modellers. In order to arrive at less speculative conclusions to inform microlensing forecasting strategies, it might be worth repeating the experiments in this chapter using a more sophisticated and realistic microlensing model.

Finally, and perhaps most significantly, this dissertation demonstrated that the combination of algorithmic intelligence with raw computing power can rival the very active and directed modelling efforts of human beings (involving the deployment of analytical techniques, the exploitation of problem-specific information, etc.), even on very difficult problems. Two decades ago, when the most powerful supercomputers in the world were no more powerful than high-end laptops of today, this would not have been the case; even a decade ago, when the cost of computing was about a thousand times higher than it is today (measured e.g. in terms of inflation-adjusted cost per GFLOPS; see Sosa, 2012), the scales would usually still have tipped in favour of human-driven modelling efforts. Today, however, intelligent search/optimisation algorithms have become a very viable alternative to developing highly-specialised approaches to solving difficult problems – a claim to which EMMA's performance bears powerful testimony.

This consideration may, admittedly, be largely academic in the context of microlensing modelling, where highly-bespoke algorithms that incorporate problem-specific information *have* already been developed and implemented – however, the consideration is applicable to *any* modelling problem, including difficult ones for which bespoke algorithms have not yet been developed.

Bibliography

Agol E., 2002, *ApJ*, 579, 430

Albrow M., et al., 1998, *ApJ*, 509, 687

Albrow M., et al., 1999a, *ApJ*, 522, 1011

Albrow M. D., et al., 1999b, *MNRAS*, 522, 1022

Alcock C., et al., 1993, *Nature*, 365, 621

Allanach B. C., Grellscheid D., Quevedo F., 2004, *J. High Energy Phys.*, 7, 69

An J. H., Albrow M. D., Beaulieu J.-P., Caldwell J. A. R., DePoy D. L., Dominik M., Gaudi B. S., Gould A., Greenhill J., Hill K., Kane S., Martin R., Menzies J., Pogge R. W., Pollard K. R., Sackett P. D., Sahu K. C., Vermaak P., Watson R., Williams A., 2002, *ApJ*, 572, 521

Asada H., 2002, *A&A*, 390, L11

Bäck T., 1996, Evolutionary Algorithms in Theory and Pratice. Oxford University Press

Bäck T., Hoffmeister F., 1990, in *Models of Selforganization in Complex Systems.* Adaptive search by evolutionary algorithms. Akademie–Verlag, pp 156–163

Bäck T., Hoffmeister F., Schwefel H., 1991, in Belew R., Booker L., eds, *Proc. 4th Int. Conf. on Genetic Algorithms.* A survey of evolution strategies. Morgan Kaufmann, pp 2–9

Baier A., Kerschbaum F., Lebzelter T., 2010, *A&A*, 516, A45+

Beaulieu J., et al., 2006, *Nature*, 439, 437

Beaulieu J. P., et al., 2010, in Coudé Du Foresto V., Gelino D. M., Ribas I., eds, *Pathways Towards Habitable Planets.* Vol. 430 of Astronomical Society of the Pacific Conference Series, EUCLID: Dark Universe Probe and Microlensing Planet Hunter. p. 266

Bennett D. P., 2008, Detection of Extrasolar Planets by Gravitational Microlensing. Springer, pp 47–88

Bennett D. P., 2010, *ApJ*, 716, 1408

Bennett D. P., Rhie S. H., 1996, *ApJ*, 472, 660

Bennett D. P., Rhie S. H., 2002, *ApJ*, 574, 985

Bevington P. R., Robinson D. K., 2003, Data reduction and error analysis for the physical sciences. McGraw-Hill

Bogdanos C., Nesseris S., 2009, *J. Cosmol. Astropart. Phys.*, 5, 6

Bond I. A., 2012, *New Astron. Rev.*, 56, 25

Bond I. A., et al., 2004, *ApJ*, 606, L155

Boss A. P., Hudgins D. M., Traub W. A., 2011, in Sozzetti A., Lattanzi M. G., Boss A. P., eds, *IAU Symposium* Vol. 276, New Worlds, New Horizons and NASA's approach to the next decade of exoplanet discoveries

Bourassa R. R., Kantowski R., Norton T. D., 1973, *ApJ*, 185, 747

Bozza V., 1999, *A&A*, 348, 311

Bozza V., 2010, *MNRAS*, 408, 2188

Bozza V., 2012, Real-time modeling of microlensing events, presented at 16th Int. Conf. on Gravitational Microlensing (available online at http://www.ipac.caltech.edu/wfir2012)

Burnham K. P., Anderson D. R., 2002, Model selection and multimodel inference: a practical information-theoretic approach. Springer

Cantó J., Curiel S., Martínez-Gómez E., 2009, *A&A*, 501, 1259

Carroll B. W., Ostlie D. A., 2006, An introduction to modern astrophysics and cosmology, 2nd edn. Benjamin Cummings

Cassan A., et al., 2012, *Nature*, 481, 167

Charbonneau P., 1995, *ApJS*, 101, 309

Charbonneau P., 2002a, Technical report, An Introduction to Genetic Algorithms for Numerical Optimization. National Center for Atmospheric Research

Charbonneau P., 2002b, Release notes for PIKAIA 1.2. National Center for Atmospheric Research

Darwin C., 1859, The Origin of Species. John Murray

Di Stefano R., Esin A. A., 1995, *ApJ*, 448, L1

Dominik M., 1998a, *A&A*, 333, L79

Dominik M., 1998b, *A&A*, 329, 361

Dominik M., 1999a, *A&A*, 341, 943

Dominik M., 1999b, *A&A*, 349, 108

Dominik M., 2004a, *MNRAS*, 353, 118

Dominik M., 2004b, *MNRAS*, 353, 69

Dominik M., 2008, in Kerrins E., Mao S., eds, *Proc. Manchester Microlensing Conf.* Modelling Microlensing Events. International School for Advanced Studies, Trieste

Dominik M., 2009, *MNRAS*, 393, 816

Dominik M., et al., 2010, *Astron. Nachr.*, 331, 671

Dong S., et al., 2009, *ApJ*, 698, 1826

Droste S., Jansen T., Wegener I., 2002, *Theor. Comput. Sci.*, 287, 131

Dyson F. W., Eddington A. S., Davidson C., 1920, *R. Soc. Lond. Philos. Trans. Ser. A*, 220, 291

Edelman A., Murakami H., 1995, *Math. Comp.*, 64, 763

Eiben A. E., Aarts E. H. L., Hee K. M. v., 1991, in *Proc. 1st Workshop on Parallel Problem Solving from Nat.* PPSN I, Global convergence of genetic algorithms: A markov chain analysis. Springer-Verlag, London, UK, pp 4–12

Einstein A., 1916, *Ann. Phys.*, 354, 769

Einstein A., 1936, *Science*, 84, 506

Fine B., Rosenberger G., 1997, The Fundamental Theorem of Algebra. Springer

Fisher R. A., 1930, The genetical theory of natural selection. Clarendon Press

Gaudi B. S., 2010, in Seager S., ed., Exoplanets. Exoplanetary Microlensing. University of Arizona Press

Gaudi B. S., 2012, *ARA&A*, 50

Gaudi B. S., et al., 2008, *Science*, 319, 927

Gaudi B. S., Gould A., 1997, *ApJ*, 486, 85

Gaudi B. S., Naber R. M., Sackett P. D., 1998, *ApJ*, 502, L33

Gaudi B. S., Petters A. O., 2002a, *ApJ*, 574, 970

Gaudi B. S., Petters A. O., 2002b, *ApJ*, 580, 468

Gaudi B. S., Sackett P. D., 2000, *ApJ*, 528, 56

Goldberg D. E., 1989, Genetic Algorithms in Search, Optimization and Machine Learning, 1st edn. Addison-Wesley Longman Publishing Co., Inc.

Goldberg D. E., Deb K., 1991, in *Foundations of Genetic Algorithms.* A comparative analysis of selection schemes used in genetic algorithms. Morgan Kaufmann, pp 69–93

Gordon V. S., Whitley D., 1993, in Forrest S., ed., *Proc. 5th Int. Conf. on Genetic Algorithms.* Serial and parallel genetic algorithms and function optimizers. Morgan Kaufmann, pp 177–183

Gould A., 1992, *ApJ*, 392, 442

Gould A., 2005, *New Astron. Rev.*, 49, 424

Gould A., 2008, *ApJ*, 681, 1593

Gould A., et al., 2010, *ApJ*, 720, 1073

Gregory P. C., 2005, Bayesian Logical Data Analysis for the Physical Sciences: A Comparative Approach with 'Mathematica' Support. Cambridge University Press

Griest K., Safizadeh N., 1998, *ApJ*, 500, 37

Haario H., Laine M., Mira A., Saksman E., 2006, *Statistics and Comput.*, 16, 339

Han C., 2006, *ApJ*, 638, 1080

Han C., 2012, Real-time Modeling, presented at 16th Int. Conf. on Gravitational Microlensing (available online at `http://www.ipac.caltech.edu/wfir2012`)

Haupt R. L., Haupt S. E., 2004, Practical Genetic Algorithms. Wiley-Interscience

Heavens A. F., Jimenez R., Lahav O., 2000, *MNRAS*, 317, 965

Hendry M. A., Bryce H. M., Valls-Gabaud D., 2002, *MNRAS*, 335, 539

Ho Y., Pepyne D., 2002, *J. Optimization Theory and Applications*, 115, 549

Hobson M. P., Efstathiou G. P., Lasenby A. N., 2006, General Relativity: An Introduction for Physicists, 1st edn. Cambridge University Press

Holland J. H., 1975, Adaptation in Natural and Artificial Systems. The University of Michigan Press

Hong T.-P., Wang H.-S., Lin W.-Y., Lee W.-Y., 2002, *Applied Intelligence*, 16, 7

Houck C. R., Joines J. A., Kay M. G., 1996, Technical report, A Genetic Algorithm for Function Optimization: A Matlab Implementation, `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.22.4413`. North Carolina State University

Hundertmark M., 2012, Characterising the Information Content of Microlensing Light Curves, presented at 16th Int. Conf. on Gravitational Microlensing (available online at `http://www.ipac.caltech.edu/wfir2012`)

Ingrosso G., Novati S. C., De Paolis F., Jetzer P., Nucita A. A., Zakharov A. F., 2009, Monthly Notices of the Royal Astronomical Society, 399, 219

Iverson K. E., 1980, *Communications ACM*, 23, 444

Jacobson N., 2009, Basic Algebra I, 2nd edn. Dover Publications

Jaroszyński M., Mao S., 2001, *MNRAS*, 325, 1546

Jenkins J. M., et al., 2010, *ApJ*, 713, L87

Kains 2012, A Bayesian algorithm for real-time model selection in caustic-crossing events, presented at 16th Int. Conf. on Gravitational Microlensing (available online at `http://www.ipac.caltech.edu/wfir2012`)

Kains N., et al., 2009, *MNRAS*, 395, 787

Kim S.-L., Park B.-G., Lee C.-U., Yuk I.-S., Han C., O'Brien T., Gould A., Lee J. W., Kim D.-J., 2010, in *SPIE Conf. Ser.* Vol. 7733, Technical specifications of the KMTNet observation system

King R. B., 1996, Beyond the Quartic Equation. Modern Birkhäuser Classics, Springer

Kochanek C. S., Keeton C. R., McLeod B. A., 2001, *ApJ*, 547, 50

Koza J., Poli R., 2005, in Burke E. K., Kendall G., eds, Introductory Tutorials in Optimization and Decision Support Techniques, Vol. 1, Search Methodologies. Springer, pp 127–164

Kramer O., 2010, *Evolutionary Intelligence*, 3, 51

Kubanek P., 2008, Master's thesis, University of Granada

Kubas D., 2005, PhD thesis, University of Potsdam

Kubas D., et al., 2005, *A&A*, 435, 941

Lagarias J. C., Reeds J. A., Wright M. H., Wright P. E., 1998, *SIAM J. Optimization*, 9, 112

Lang S., 1993, Real and Functional Analysis. Springer

Li M., Vitanyi P., 1997, An introduction to Kolmogorov complexity and its applications, 2nd edn. Springer-Verlag

Liebes S., 1964, *Phys. Rev.*, 133, 835

Liebig C., Wambsganss J., 2010, *A&A*, 520, A68

Liesenborgs J., De Rijcke S., Dejonghe H., 2006, *MNRAS*, 367, 1209

Lim D., Ong Y.-S., Jin Y., Sendhoff B., Lee B.-S., 2007, *Future Generation Comput. Syst.*, 23, 658

Lloyd S., 2002, *Phys. Rev. Lett.*, 88, 237901

Maitre O., Baumes L. A., Lachiche N., Corma A., Collet P., 2009, in *Proc. 11th Annu. Conf. on Genetic and Evolutionary Comput.* GECCO '09, Coarse grain parallelization of evolutionary algorithms on GPGPU cards with EASEA. ACM, pp 1403–1410

Mao S., 2008, in Kerrins E., Mao S., eds, *Proc. Manchester Microlensing Conf.* Introduction to gravitational microlensing. International School for Advanced Studies, Trieste

Mao S., 2012, *Res. Astron. Astrophys.*, 12, 1

Mao S., Di Stefano R., 1995, *ApJ*, 440, 22

Mao S., Paczyński B., 1991, *ApJ*, 374, L37

Mason J., ed. 2008, Exoplanets. Praxis Publishing Ltd

Mayr E., 1963, Animal species and their evolution. Belknap Press

Metcalfe T. S., Nather R. E., Winget D. E., 2000, *ApJ*, 545, 974

Michalewicz Z., 1996, Genetic Algoritms + Data Structures = Evolution Programs. Springer

Michalewicz Z., Fogel D. B., 2000, How to Solve It: Modern Heuristics, 2nd edn. Springer

Miller B. L., Goldberg D. E., 1995, *Complex Syst.*, 9, 193

Misner C. W., Thorne K. S., Wheeler J. A., 1973, Gravitation, 2nd edn. W. H. Freeman

Mokiem M. R., de Koter A., Puls J., Herrero A., Najarro F., Villamariz M. R., 2005, *A&A*, 441, 711

Molga M., Smutnicki C., 2005, Test functions for optimization needs, available online at `http://zsd.iiar.pwr.wroc.pl/files/docs/functions.pdf`

Moré J. J., Garbow B. S., Hillstrom K. E., 1981, *ACM Trans. Math. Softw.*, 7, 17

Nelder J. A., Mead R., 1965, *Comput. J.*, 7, 308

Nesseris S., Shafieloo A., 2010, *MNRAS*, 408, 1879

Night C., Di Stefano R., Schwamb M., 2008, *ApJ*, 686, 785

Paczyński B., 1986, *ApJ*, 304, 1

Paczyński B., 1996, *ARA&A*, 34, 419

Pardalos P. M., Rosen J. B., 1987, Constrained global optimization: algorithms and applications. Springer-Verlag

Petters A. O., Levine H., Wambsganss J., 2001, Singularity theory and gravitational lensing. Birkhäuser

Pospichal P., Jaros J., Schwarz J., 2010, in Chio D., et al. eds, Lecture Notes in Computer Science, Vol. 6024, Applications of Evolutionary Computation. Springer, pp 442–451

Potter M., De Jong K., 1994, in Davidor Y., Schwefel H.-P., Mnner R., eds, Lecture Notes in Computer Science, Vol. 866, Parallel Problem Solving from Nature III. Springer, pp 249–257

Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P., 2007, Numerical Recipes: The Art of Scientific Computing, 3rd edn. Cambridge University Press

Quanz S. P., Lafrenière D., Meyer M. R., Reggiani M. M., Buenzli E., 2012, *A&A*, 541, A133

Rajpaul V., 2011, in Basson I., Botha A. E., eds, *Proc. SAIP 2011.* Genetic algorithms in astronomy and astrophysics. UNISA, pp 519–524

Rajpaul V., 2012, *MNRAS*, 427, 1755

Ralston A., Rabinowitz P., 2001, A First Course in Numerical Analysis, 2nd revised edn. Dover Publications

Refsdal S., 1964, *MNRAS*, 128, 295

Refsdal S., 1966, *MNRAS*, 134, 315

Risco-Martn J., Lanchares J., Coello-Coello C., 2012, *Soft Comput.*, 16, 185

Romanov V. G., 1987, Inverse problems of mathematical physics. VSP International Science

Ryu Y.-H., Chang H.-Y., Park M.-G., 2011, *MNRAS*, 412, 503

Schneider P., Ehlers J., Falco E. E., 1992, Gravitational Lenses. Springer

Schneider P., Kochanek C. S., Wambsganss J., 2006, Gravitational Lensing: Strong, Weak and Micro. Springer

Schwarzschild K., 1916, in *Sitzungsberichte der Deutschen Akademie der Wissenschaften zu Berlin, Klasse fur Mathematik, Physik, und Technik.* Über das Gravitationsfeld eines Massenpunktes nach der Einsteinschen Theorie

Sharma G., Martin J., 2009, *Int. J. Parallel Program.*, 37, 3

Shin I.-G., et al., 2011, *ApJ*, 735, 85

Shin I.-G., et al., 2012, Characterizing Low-Mass Binaries From Observation of Long Time-scale Caustic-crossing Gravitational Microlensing Events, preprint, available online at `http://arxiv.org/abs/1204.2869`

Shvartzvald Y., Maoz D., 2012, *MNRAS*, 419, 3631

Skowron J., et al., 2011, *ApJ*, 738, 87

Sosa C., 2012, Introduction to High-Performance Computing, presented at the University of Minnesota Rochester (available online at `http://www.msi.umn.edu/~cpsosa`)

Srinivas M., Patnaik L. M., 1994, *Comput. J.*, 27, 17

Stevens J. B., 2005, PhD thesis, The University of Melbourne

Stewart J., 2008, Calculus: Early Transcendentals. Thompson Brooks/Cole

Storn R., Price K., 1997, *J. Global Optimization*, 11, 341

Sumi T., 2011, Models for MOA-2011-BLG-197/OGLE-2011-BLG-0265, available online at `http://www.stelab.nagoya-u.ac.jp/~sumi/anomaly/2011/MB11197.html`

Sumi T., et al., 2003, *ApJ*, 591, 204

Sumi T., et al., 2010, *ApJ*, 710, 1641

Sumi T., et al., 2011, *Nature*, 473, 349

Syswerda G., 1989, in Schaffer D. J., ed., *Proc. 3rd Int. Conf. on Genetic Algorithms.* Morgan Kaufmann Publishers, Inc., pp 2–9

Tarantola A., Valette B., 1982, *J. Geophys.*, 50, 150

Thierens D., Goldberg D., 1994, in Davidor Y., Schwefel H.-P., Mnner R., eds, Lecture Notes in Computer Science, Vol. 866, Parallel Problem Solving from Nature III. Springer, pp 119–129

Tsapras Y., et al., 2009, *Astron. Nachr.*, 330, 4

Tsutsui S., Ghosh A., Corne D., Fujimoto Y., 1997, in *Proc. 7th ICGA*. A real coded genetic algorithm with an explorer and an exploiter populations. Morgan Kaufmann, pp 238–245

Udalski A., 2003, *Acta Astronomica*, 53, 291

Udalski A., Kubiak M., Szymanski M., 1997, *Acta Astronomica*, 47, 319

Udry S., Santos N. C., 2007, *ARA&A*, 45, 397

Veller C., Rajpaul V., 2012, *Phys. Rev. E*, 86, 041907

Vermaak P., 2000, *MNRAS*, 319, 1011

Vermaak P., 2003, *MNRAS*, 344, 651

Vermaak P., 2007, PhD thesis, University of Cape Town

Wahde M., Donner K. J., 2001, *A&A*, 379, 115

Walsh D., Carswell R. F., Weymann R. J., 1979, *Nature*, 279, 381

Wambsganss J., 2011, *Nature*, 473, 289

Wehrens R., Buydens L. M. C., 1998, *Trends in Analytical Chemistry*, 17, 193

Witt H. J., 1990, *A&A*, 236, 311

Witt H. J., 1993, *ApJ*, 403, 530

Witt H. J., Mao S., 1994, *ApJ*, 430, 505

Wolpert D., Macready W., 2005, *Evolutionary Comput.*, 9, 721

Wolpert D. H., Macready W. G., 1997, *IEEE Trans. Evol. Comp.*, 1, 67

Woźniak P. R., 2000, *Acta Astronomica*, 50, 421

Wright A. H., 1991, in Rawlins G. J., ed., *Foundations of genetic algorithms.* Genetic Algorithms for Real Parameter Optimization. Morgan Kaufmann, pp 205–218

Wright A. H., Vose M. D., Rowe J. E., 2003, in *Proc. 2003 Int. Conf. on Genetic and Evolutionary Comput.* Springer-Verlag, pp 1505–1517

Zakharov A. F., 1995, *A&A*, 293, 1

Zwicky F., 1937, *Phys. Rev.*, 51, 290