CHAPTER 1

AUTOMATIC FITTING OF TYPE IA SUPERNOVA SPECTRA

1.1. Introduction

The last chapters (Chapters **??**) were dedicated to the hunt for donor stars and did not use the measurements from the Type Ia supernovae (henceforth SNe Ia) themselves. In this chapter we will describe the extraction of nucleosynthetic yields and energies from optical spectra as well as the automation of this process.

The main sources of information in spectra are the shape/strength of the spectral features and their evolution with time. There have been a few attempts to extract the details of the stellar explosions from one or both of these sources. All of them employ the technique of fitting the observations using synthetic spectra. A critical part of this process is the radiative transfer algorithm used to generate the synthetic spectra, which has been implemented in several different codes.

Fisher (2000) wrote a very simple radiative transfer code called SYNOW. SYNOW is a highly parametrized code and thus is mainly used for line identification rather than actual fitting of SN Ia spectra. The main code used in this work is a further development of the code described in Mazzali & Lucy (1993) and Mazzali (2000), henceforth named the 'Mazzali and Lucy Monte Carlo' code (MLMC). Compared to the SYNOW code the MLMC code calculates a radiative equilibrium temperature and uses this to compute internally consistent ionization ratios. In addition, MLMC takes electron scattering into account as well as allowing for non-resonance line scattering.

Codes such as PHOENIX Hauschildt & Baron (1999), SEDONA Kasen et al. (2006) and ARTIS Kromer & Sim (2009) are powerful 3D radiative transfer codes. They are the most 'physical' codes available but take hours to days on supercomputers to produce spectra. As they take a considerable amount of time and computational power, these codes, however, are not practical for fitting many observed spectra.

Mazzali et al. (2007) showed that the brightness of a supernova relies mainly on the production of ^{56}Ni. Faint SNe Ia (91bg-like) produce roughly $0.2\,M_{\odot}$of ^{56}Ni. Most SNe Ia produce roughly $0.6\,M_{\odot}$, with some luminous SNe Ia (91T-like) producing up to $0.9\,M_{\odot}$. All luminosity types enclose approximately $1\,M_{\odot}$ of ash. This remarkable find was made by manually fitting 23 SNe Ia using the MLMC. There are however many more spectroscopically well sampled SNe Ia and soon there will be even more observed expected from the next generation supernova searches. Such large datasets cannot be effectively studied with manual fitting techniques. An automated procedure can not only fit many supernovae but will also explore the parameter space to calculate error estimates. Strict quality measures necessary for an automatic fitter can also help when interpreting physical quantities generated by the code. We present an automated way of fitting supernovae using the MLMC (in collaboration with Stephan Hachinger and Paolo Mazzali).

In section 1.2 we will introduce the inner workings of the MLMC-code. We will discuss the properties of the parameter space on one example fit in Section 1.3. Section 1.4 provides an introduction to GA, which are the optimization strategies of choice in the automatization of the MLMC. We will present the current implementation of the autofitting code in Section **??** (Dalek code). Finally we will conclude and give an outlook over future work of this project in Section 1.6.

1.2. The MLMC Code

The MLMC code, used in this work, is described in detail in Mazzali (2000) and we will give only a very short introduction to the key concepts here. We refer the reader to Mazzali (2000) for a more detailed description of the code. A more advanced version, using abundance stratification, but not used in this work is described in Mazzali (2000).

Based on spectra, the phase of the evolution of a supernova can be divided into the photospheric phase and the nebular phase. The MLMC only creates synthetic spectra for SN Ia in the photospheric phase. In the photospheric phase the supernova can be approximated by a sharp photosphere emitting a black-body spectrum with a fast moving ejecta-layer on top.

1.2.1. Radiative Transfer

Radiative transfer calculations attempt to provide the wavelength dependent flux emerging from the atmosphere for a given input boundary condition at the base of the atmosphere. Computing the attenuation for the given input flux (F_0) is a critical part of this process. This wavelength-dependant attenuation factor is called the opacity τ:

$$F(\lambda) = F_0(\lambda)\, e^{-\tau(\lambda)}, \tag{1.1}$$

where F is the observed flux and F_0 is an assumed distribution of input flux before being absorbed by the plasma, which imposes the attenuation factor $e^{-\tau}$. There are many physical processes relevant to supernova radiative transfer. The line opacity (bound-bound transitions) has the biggest impact on the final spectrum. The lines in SN Ia spectra are the prime indicators for elemental abundance. In contrast to lines, which occur at one specific frequency, Thompson scattering is independent of frequency and only depends on the state of the atmosphere. Thompson scattering is an important way to redistribute photons in angular space (i.e. making the radiation field more isotropic) and thus also has a strong influence on radiative heating/ionization of the plasma. Other sources of opacity like free-free absorption and bound-free absorption are thought of as second order effects and are not implemented in the MLMC. As the MLMC is required to be fast only bound-bound opacity and Thompson scattering is implemented in the code.

Unlike stellar atmospheres, in supernova ejecta one needs to consider the photon's Doppler shift in relation to the surrounding medium. One major assumption that the code makes is the so-called Sobolev approximation (Sobolev, 1960). The Sobolev approximation assumes that lines have only a single frequency at which they can be excited (no natural, thermal or other line broadening), This assumptions is a good approximation where photons are moving through a medium where the velocity gradient is so high that they are only in resonance with one line (no blending) in a very small region. This Sobolev approximation is one important factor of making the code fast yet still reproducing the observed spectra rather well.

In addition the MLMC assumes the ejecta to be in homologous expansion. This means that the velocity is a linear function of the radius:

$$v = r/t.$$

Combining both the Sobolev approximation with the assumption of homologous expansion yields this relatively simple formula for line opacities:

$$\tau_{lu} = \frac{\pi e^2}{m_e c} f \lambda t_{\mathrm{exp}} n_l \left(1 - \frac{g_l n_u}{g_u n_l}\right), \tag{1.2}$$

where τ_{lu} denotes the opacity going from the lower state to the upper state of an atom, e is the electron charge, m_e is the electron mass, f is the absorption oscillator strength of the line, λ denotes the wavelength, t_{exp} the time since explosion, n_x the number of atoms in the state x and g_x is the statistical weight of the state x.

Although producing relatively good synthetic spectra, the assumptions of homologous expansion and Sobolev approximation have their caveats. In the case of homologous expansion it is thought to be a very good approximations after the first few minutes of the explosion, but maybe effected when, for example, the ejecta interacts with surrounding material. The main caveat for Sobolev approximation is that a line is not a delta-function, as assumed in the Sobolev approximation. If two strong lines are close in frequency space it can lead to the first line shielding the second line. Despite these caveats, however, the Sobolev approximation and homologous expansion are widely used and have been demonstrated to provide an adequate compromise between accuracy and computational practicality for fitting the spectra of supernova ejecta.

We have discussed the propagation of the photons in the plasma but have not discussed the state of the plasma yet. The simplest assumption for the state one can make is *local thermodynamic equilibrium* (LTE). In this case the Boltzmann formula describes the level populations in a single ion:

$$\frac{n_j}{n_{\mathrm{ground}}} = \frac{g_j}{g_{\mathrm{ground}}} e^{-(\epsilon_j - \epsilon_{\mathrm{ground}})/kT},$$

where n_j is the population of the j-th state of the atom, g_j is the statistical weight of the j-th state, ϵ_j is the energy of the j-th state and T is the temperature of the plasma. Similarly we can calculate the ionization state using the Saha-equation:

$$\frac{N_j}{N_{j+1}} = n_e \frac{U_j(T)}{U_{j+1}(T)} C_I T^{-3/2} e^{\chi/kT},$$

where N_x are the total ion population with ionization state x, U_x is the partition function for the ionization state x, n_e is the number density of electrons, C_I is a universal constant and all other symbols have their usual meaning.

In the nebular approximation (Mihalas, 1978) we approximate the radiation field by a diluted black body:

$$J = WB(T_R),$$

where J is the mean intensity, T_R is the radiation temperature and W is the dilution factor. In the standard nebular approximation W differs from 1.0 due to purely geometrical reasons: the photon count falls with distance to the source ($1/r^2$ for a point source). For an extended source the geometric part of the dilution factor takes the following form (Mihalas, 1978):

$$W(r) = \frac{1}{2}\left[1 - \sqrt{1 - \left(\frac{r_{\mathrm{ph}}}{r}\right)^2}\right] \tag{1.3}$$

This concept can be generalised to a modified nebular approximation where the effects of attenuation and scattering (e.g. line and Thompson scattering) on W can also be included. Using W and J and assuming LTE one can easily calculate the electronic and ionization states of the plasma:

$$\frac{n_j}{n_{\mathrm{ground}}} = W\left(\frac{n_j}{n_{\mathrm{ground}}}\right)_{T_R}^{\mathrm{LTE}}$$

and

$$\frac{N_j}{N_{j+1}n_e} = W\left(\frac{N_j}{N_{j+1}n_e}\right)_{T_R}^{\mathrm{LTE}}.$$

This simple form of the nebular approximation only considers radiative excitation/de-excitation and ionization/recombination from and to the ground state, but has the advantage of solving the state of the plasma computationally inexpensively. The MLMC uses a variant of the modified nebular approximation which includes radiative excitation/de-excitation from excited states and a different treatment of the ultra violet (UV) radiation field (which contains most ionization edges). For a detailed description of the modifications please refer to Mazzali & Lucy (1993) and references therein.

In the MLMC code the two fundamental properties of the radiation field - temperature and the dilution factor - are determined from a *Monte Carlo* (MC) simulation. We first measure the mean frequency of the photons and bolometric intensity. We then infer the temperature by requiring the black body to have the same mean frequency (Wien approximation). The dilution factor can then be fitted to match the bolometric intensity. We note that close to the photosphere the dilution factor is influenced mainly by the geometry and should be close to 0.5 (see Equation 1.3 for $r = r_{\mathrm{ph}}$). As the MC process is statistical and the dilution factor an approximation this might vary, but we assume this to be still an important factor to determine if the model is realistic.

In summary, the approximations described in this section make the MLMC fast, yet still providing an acceptable representation of reality. This makes the MLMC code feasible to use for modelling data.

1.2.2. Monte Carlo Radiative Transfer

We have so far only described the approximations made by the code, but not how these are implemented and work together. In this section we will give an overview of the process and design of the MLMC code.

First the ejecta in the MLMC is divided into 20 concentric shells with an equal thickness in $1/r$, where r is the radius from the centre of the explosion. Each shell has a uniform density which is drawn from the well known empirical W7 model (Nomoto et al., 1984). The MLMC also allows the change of this density structure, but for all this work we have kept the W7 model density structure. In addition, we assume a homogenous abundance distribution throughout the whole model. There is, however a *stratified* version of the code that allows for different abundances in each shell. We have opted to use the simple homogeneous version as the parameter space is very complex even for this homogeneous version. We plan to extend the automatic fitting procedure to use the stratified version at a later date. In addition, we calculate the time since explosion t_{exp} using the time of the photometric maximum and adpopting a rise time estimate of 19.5 days. The photospheric velocity v_{ph}, the bolometric luminosity L_{bol} and abundances for the chosen elements are the input parameters to the MLMC.

To conserve radiative equilibrium the MLMC uses photon packets instead of individual photons. Each photon packet, described by frequency and number of photons, contains the same energy (more photons per packet in the red than in the blue). For pure elastic scattering events - the absorption frequency is the same as the reemission frequency - the number of photons is conserved in each packet. For line absorptions this is generally not the case. The MLMC allows for photon branching, which means that the photon can be emitted in a different transition than it was absorbed in. Once absorbed the new emission transition is chosen by a weighted random process (for a more details see Mazzali, 2000). The number of photons in the packet is adjusted to conserve the co-moving energy of the packet, thus preserving the energy between matter and radiation field (radiative equilibrium).

The MC simulation begins by calculating the plasma condition using an initial guess of temperature and dilution factor for each shell. A photon packet is emitted with a random frequency and a random angle drawn from a blackbody distribution. An event optical depth is calculated from a uniform random distribution so that $\tau_{\mathrm{event}} = -ln(z)$, $z \in (0, 1]$ (follows from sampling the Equation 1.1). There are three possible outcomes for the photon in each Monte Carlo step. First we calculate the length of the path (s_e) that the packet can travel freely before τ_{event} is equal to the Thompson scattering opacity $\tau_{\mathrm{event}} = \sigma_{\mathrm{T}} n_e s_e$. Secondly we calculate the same path length for the lines s_l using as a target opacity $\tau_e + \tau_{\mathrm{line}}$ for line scattering (where $\tau_e = \sigma_{\mathrm{T}} n_e s_l$ and τ_{line} is the line opacity calculated with Equation 1.2). If s_e is the shortest then Thompson scattering occurs and the photon is assigned a new direction and a new τ_{event} is drawn and we start anew. If $s_e < s_l < s_{\mathrm{exit}}$, where s_{exit} is the length to exit the shell, line scattering occurs and the photon is absorbed by an atom. This excited atom can then de-excite through many lines. MLMC randomly chooses a downward transition for the whole packet (taking the appropriate weights into account) and adjusts the photon number in the packet to ensure radiative equilibrium. Finally, if both paths are longer than the path to exit the current shell, then the photon exits the current zone and a new MC step begins.This process is iterated until either the photon packet escapes the outermost shell - in which case the photon number and frequency is recorded - or penetrates the photosphere in which case the photon is discarded and a new one drawn.

This iterative process is first used in the MLMC code to prepare the plasma state. In this initial simulation each packet status is recorded at the mid-point of each shell. This information is used to calculate a new temperature and dilution factor which then leads to updated plasma conditions (level populations and ionizations). This procedure is repeated until the plasma temperature converges. Once convergence is reached the actual Monte-Carlo simulation begins.

The final spectrum is not calculated using the number and frequency of the escaping photon packets. Instead we use the photon packets to estimate the source function as a function of wavelength and then calculate the emerging spectrum using a formal integral solution of the radiative transfer equation (a more detailed description of this method in Mazzali, 2000). This has the advantage of reducing noise in the spectrum due to MC noise.

In summary, the MLMC produces realistic spectra in a competitive calculation time (one spectrum per minute on a modern computer). A more detailed description of the code can be found in Mazzali & Lucy (1993) and Mazzali (2000). There is a abundance stratified version of the MLMC code, which has not been described in this work but can be reviewed in **?**.

1.3. Manually fitting a Type Ia supernova

We extract physical quantities from the SN Ia spectra by adjusting the parameters of the MLMC until the code produces a similar synthetic spectrum to the observed one. For now this fitting process for SNe Ia is an arduous manual task, which requires detailed knowledge about the inner workings of the MLMC and its approximations. Known problems, like the excess in the near infrared (NIR) (described in more detail later) and the high velocity calcium, necessitate one to prioritize between different quality measures. This also leads to a subjective 'best-fit', although experts in MLMC fitting seem to come to the same conclusion despite their different approaches. We will first illustrate this fitting process on one example - SN 2002bo 10 days before maximum (Benetti et al., 2004) - which has been fitted manually by Hachinger (2007), before moving to our automatic optimization attempts in Section 1.5.

The first step in any fitting procedure is to initialize the parameters. The luminosity distance, redshift and time since explosion t_{exp} are initialized from data available on the supernova and are not fitted with the MLMC code. The initial choice of luminosity and photospheric velocity is based empirically on photometry and t_{exp}. When manually fitting the spectrum we only consider contributions from the elements listed in Table 1.1. When initializing these for a manually fitting procedure we use empirical data (this procedure is described in detail in Section 1.5). In the fitting process several elements are treated specially. The abundance of ^{56}Ni for example, is given as the abundance of initial nickel synthesized in the explosion. Since ^{56}Ni is unstable this unified mass of ^{56}Ni will decay first to ^{56}Co which then decays to ^{56}Fe. This decay is accounted for (i.e. the abundances of ^{56}Ni, ^{56}Co and ^{56}Fe are calculated from the initial ^{56}Ni mass and t_{exp}). In all further discussions, cobalt will always refer to radioactive ^{56}Co, the daughter of ^{56}Ni (we do not consider any stable isotopes of cobalt that may have been synthesized in the explosion). The ^{56}Fe abundance is determined by adding the ^{56}Fe obtained from the ^{56}Ni decay to the

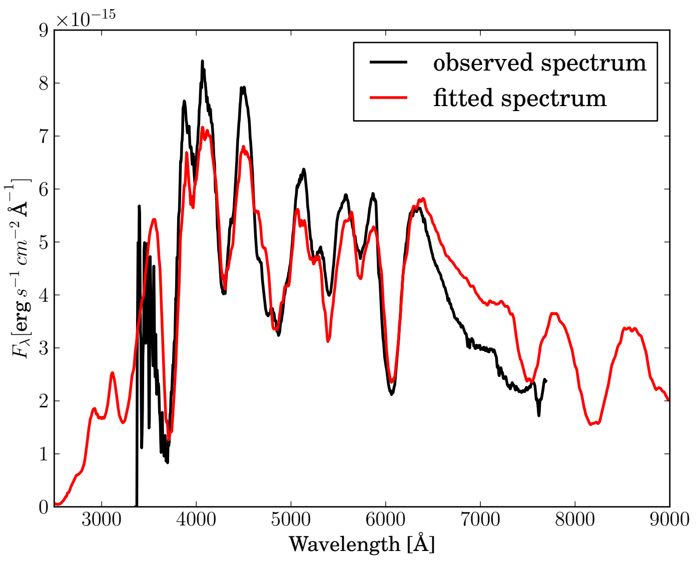

Figure 1.1 Spectrum of SN 2002bo (Benetti et al., 2004) with MLMC-fit by Hachinger (2007). The excess in redwards of 6500 Å is a common problematic features of these fits.

iron and pre-exisiting ^{56}Fe (Fe$_0$). In addition we lock the ratio of titanium and chromium as we do believe this to be degenerate in the fit.

explosive nucleosynthesis reference (e.g. Iwamoto et al., 1999)

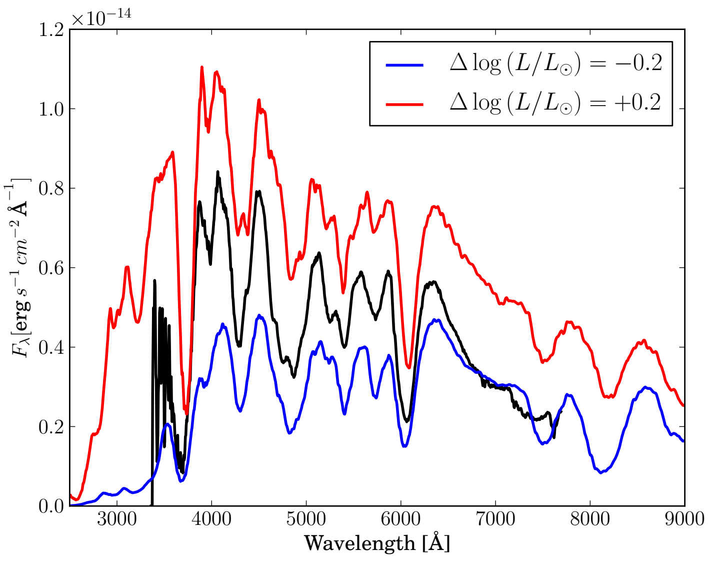

Figure 1.2 We have perturbed the luminosity around the best fit value. The most noticeable effect is the continuum offset. There is also a slight change in the overall slope of the spectrum.

Next we will describe the unique impact each parameter has on the spectrum and how that is used to arrive at a best fit. There are three main parameters that have the most influence on the overall spectrum fit: Luminosity, photospheric velocity and abundance in iron group element (IGE). A large offset in L, relative to the best fit parameter, is easily

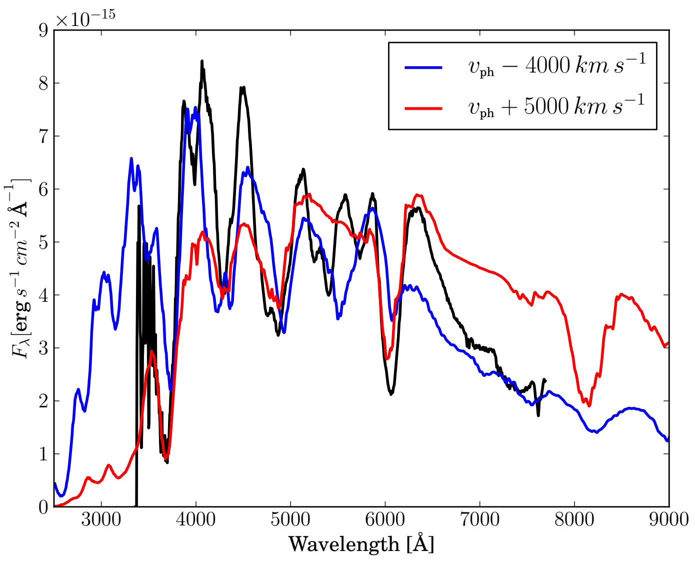

Figure 1.3 The photospheric velocity has been perturbed around the bestfit. If the velocity is too small, the spectrum is too blue and vice versa.

visible as a large offset of the continuum (see Figure 1.2). Thus it is easy to constrain the parameter space in L. L also has influence on the temperature of the model through:

$$L_{\rm bol} = 4\pi\sigma R^2 T_{\rm eff}^4 = 4\pi\sigma(v_{\rm ph}t_{\rm exp})^2 {T_{\rm eff}}^4,$$

where $T_{\rm eff}$ can be calculated by the model parameters L and $v_{\rm ph}$.

Velocity in astronomy is often measured using the Doppler shift of atomic lines. In this case however it is hard to measure $v_{\rm ph}$ from atomic lines. Lines are created at different depths and thus at different velocities. This smears out the line profiles which makes fitting velocities nearly impossible using this technique. The main impact of photospheric velocity is establishing the temperature structure with the given luminosity. A model with a too high photospheric velocity will have expanded more than the observed spectrum and thus will be cooler. This results in a spectrum that is too luminous in the red and not luminous enough in the blue (see Figure 1.3). A secondary effect is that the ionization state will be wrong (blue radiation field will lead to more photoionization and vice versa), which can be seen in various lines.

The IGE have a similar influence on the overall flux distribution as the photospheric velocity. We assume, as discussed previously, no stable cobalt. All IGE elements reprocess the flux heavily by absorbing bluewards of $\approx$ 3800 Å and fluoresce in the red part of the spectrum. For example, a too high abundance will suppress the flux in the blue too much and will cause the spectrum to be over-luminous in the red (see Figure **??**). Although physically different from the photospheric velocity, phenomenologically these are similar. The degeneracy is broken by identifiable iron lines in the spectrum as well as the ionization balance determined by the temperature (mainly influenced by photospheric velocity and luminosity). This near degeneracy causes a very complex parameter space. The titanium/chromium abundance ratio is fixed and only one of these elements is fit. Like Ni_0 and Fe_0, titanium and chromium provide strong flux suppression in the blue and fluoresce in the red. Chromium can be partly constrained due to a blended line just blueward of the

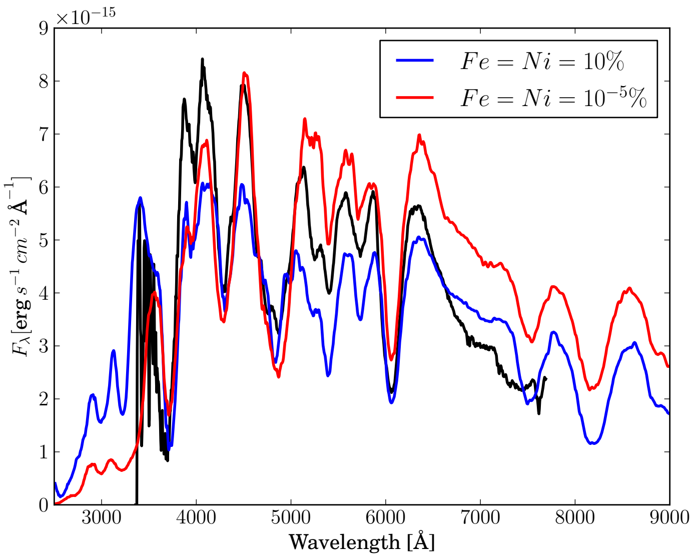

Figure 1.4 When changing the IGE (in this case we have only changed Fe_0 and Ni_0) the flux is altered in the blue and red part. Too much IGEs and there's not enough flux in the blue and too much flux in the red and vice versa.

UV calcium feature, but other than that we believe them to be slightly degenerate with the other IGE. We disregard the elements scandium, vanadium and manganese as they seem to have little influence on the spectrum at their predicted abundances.

There are six other abundances that are taken into account when fitting: carbon, oxygen, magnesium, silicon, sulphur and calcium (see Figure 1.5). Judging the fit of the Ca ii line at 3700 Å is relatively easy and thus the calcium abundance is usually changed first. An additional constrain on the calcium abundance is the NIR triplet near 8500 Å, which is however often missing in spectral coverage. The choice of temperature imposed by photospheric velocity and luminosity does not have an immense influence on this line. One caveat however is that the Ca ii line saturates at a certain abundance. If the observed Ca ii line is close to that limit one can only extract a lower limit for the calcium abundance.

Silicon and sulphur are usually the next elements to be fine-tuned. Both of these elements are linked through nuclear synthesis and we do not expect there to be more sulphur than silicon. We also expect no less sulphur than a third of silicon. Silicon also provides an important measure for temperature through the ionization balance, which is seen as the ratio between Si ii at 6150 Å versus the fit of Si ii at 5700 Å (see Figure 1.3). This effect was first mentioned in **?** and is explained in detail in (**?**). The strong Mg ii feature near 4300 Å helps constrains the magnesium abundance. The weak C ii feature at ≈ 6300 Å is the only line to provide constraints for the carbon abundance. In most SNe Ia spectra this line is weak or not visible and mainly provides an upper limit for the amount of carbon.

The last element to be described is oxygen, which takes the rest of the mass fraction when all other elements have been assigned abundance fractions. We will sometimes refer to this as a buffer element, but we caution the reader to not think of this as a physical description but rather a description of implementation. Such a buffer is needed as we have chosen a density distribution (W7 model) and the elemental abundances are fractions by mass. Oxygen is chosen for that role as it does not have strong influence over the overall

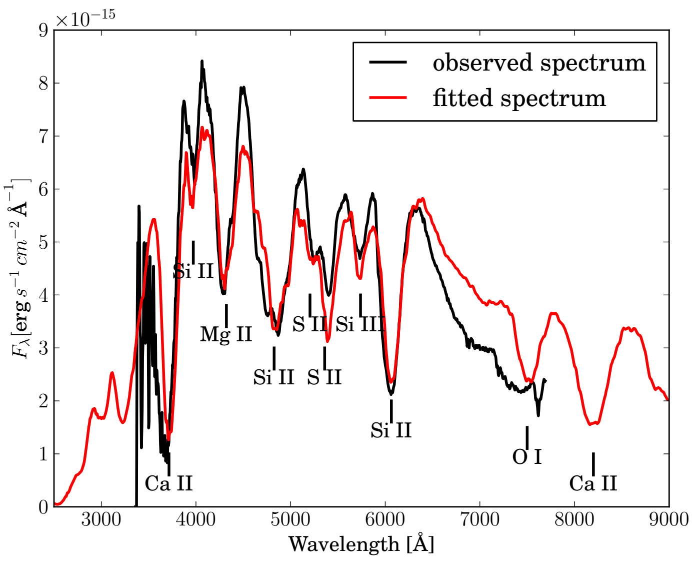

Figure 1.5 The best-fit model for SN 2002bo We have identified some of the strongest lines that are scrutinized when fitting a SN Ia spectrum.

spectrum. Oxygen in general is not a major contributor to lines, the only strong lines are the O I triplet at 7774 Å. This line is often very weak as a large fraction of oxygen is ionized and significant contamination from silicon and magnesium to the triplet. Both of these arguments mean that large fractions of oxygen might not be visible in the spectrum making it a perfect buffer element.

The fitting process usually involves first adjusting luminosity, then IGEs and finally photospheric velocity. This is followed by adjusting the other elemental abundances from the initial values. After the elemental abundances are adjusted we re-adjust the luminosity, photospheric velocity and IGE. This iterative process continues until a satisfactory fit has been obtained. When closing in on the optimal fit we consider the dilution factor. Purely theoretical we would expect this to be close to 0.5 for a 'physical' fit. We do, however, accept values between 0.4 and 0.7 as physical. This large range is accepted due to numerical fluctuation and approximations made by the MLMC. The parameters are often improved further by checking their time evolution through fits of photospheric spectra at different times.

Finally, during and after the fitting process the elemental abundances and ratios (listed in Table 1.1) are also checked against theoretical nucleosynthetic yields (e.g. Iwamoto et al., 1999). If the abundances and abundance ratios are outside of these generous bounds one normally tries to find another reasonable fit within the bounds. We have also implemented this checking into our automatic optimization routines (see Section 1.5).

In summary, the fit is relatively good as seen in Figure 1.5. There are however some problems that are intrinsic to the MLMC and can not be rectified by adjusting parameters. The calcium line at 3900 Å can be seen to be to blueshifted in relation to the model. This property is not unusual and is thought to come from high velocity calcium components at the outer edge of the ejecta. This can often be rectified in the stratified version of the MLMC which is not discussed in this work. The next major known discrepancy is the excess of flux redwards of ≈ 6200 Å. This is a common problem rising assumption of an

Table 1.1 Parameters for best fit

Parameter	Value	Realistic Constrains
$\log L/L_\odot$	9.05	
v_{ph}	11700	
Carbon	0.08 %	$C < 12.5\,\%$
Oxygen	54.9 %	$C < O$
Magnesium	10 %	-
Silicon	25 %	$Si > 1\,\%$
Sulphur	4.5 %	$1 < Si/S$-ratio < 10
Calcium	1 %	$Ca < 5\,\%$
Titanium	0.01 %	$Ti + Cr < 1\,\%$
Chromium	0.07 %	Cr/Ni_0-ratio < 10
pre-existing Iron	0.07 %	Fe_0/Ni_0-ratio < 10
undecayed Nickel	1.5 %	$Ni_0 < 80\,\%$

underlying black body spectrum, which overestimates the flux in this region. Although the continuum is offset we still believe the line depth to be representative, thus when fitting manually often one tries to fit the depth of the lines instead of the continuum. This excess makes it difficult to use traditional fitness measures and we will discuss the path we took in Section 1.5.

In the end, the fitting of a supernova is a complex procedure and requires a lot of practice. For automating this process we initially tested simple gradient methods. These failed abysmally. We quickly discovered the search space is too complex and evaluation time takes too long (each synthetic spectrum roughly one minute on a modern computer) to use these simple methods. Research in numerical optimization techniques has made significant process in the last decades. These areas of mathematics have yielded impressive algorithms. Genetic algorithms are easy to implement and are intrinsically parallel. A perfect match for our problem.

1.4. Brief Introduction to Genetic Algorithms

Before advancing to the description of the automation of the MLMC we will give a brief introduction to GAs which are used in the automation endeavour. We urge the reader to review GAs in Chapter **??** before proceeding.

The complexity of an optimization problem rises with the number of dimensions, but more so with the correlation of the input parameters. Iterative algorithms, often using gradients to find extrema by solving a Hessian matrix, are very good at problems with a low number of dimensions and small correlation between input parameters. They can exactly solve the problem in a very small amount of computational time. However as the search space gets more complex and multiple extrema start to appear. The iterative algorithm often converge on those and are only useful when starting close to the global optimum. For these complex multi-dimensional parameter spaces stochastic optimization techniques that rely heavily on random numbers are often the best choice. Unlike the iterative methods, however, stochastic optimization algorithms are not guaranteed to find any optimum.

Automatically fitting SN Ia spectra is a very complex multi-dimensional parameter space with many highly correlated parameters. We have chosen GAs - a stochastic optimization algorithm - to conquer this problem. GAs use terminology which borrows from its roots in evolutionary science. At the core of each GA is the individual. The individual represents one solution in search space. The genotype of the individual is the vector representation of all of its parameters, sometimes referred to as genome. Evaluating the genotype leads to the phenotype which is the individual's representation in solution space (e.g. a spectrum obtained from parameters). For optimization one needs to have a quality measure which in terms of GAs is the fitness. The fitness is calculated using a fitness function from the phenotype of the individual (e.g. the χ^2 value calculated for a synthetic spectrum against the observed spectrum).

GA have a number (called population size; as a rule of thumb ten times the number of parameters) of these individuals in a collection known as a population or generation. This enables them to probe multiple areas in search space simultaneously, which makes these algorithms intrinsically parallel. During the optimization process GAs go through several stages first of which is the initialization of the first generation. The easiest is to draw random parameters from the parameter space (there are more advanced techniques which are described in Chapter **??**). Once this first generation is populated with the required number of individuals we evaluate their phenotypes and then their fitness. Once this step is complete we want to form the second generation out of the old generation. Chapter **??** gives a very detailed overview over the different processes building a new population, we will here only describe one path. First we use a method of randomly choosing individuals that still favours more fit individuals over less fit individuals. Using this process we select two individuals of the old population for mating. For the mating process we employ a technique called single-point crossover to create a new individual. This means that we select a random point in the genome of both parents. Parameters before this point are taken from the first parent, parameters after that point are taken from the second parent. This newly created individual (or child) is then subjected to possible mutation, where each parameter has a small chance of being multiplied by a random number. We repeat this process of choosing two parents and creating children until the new generation has the same number of individuals as the old population. One should note that in this process it is possible, unlike in nature, that the parents are the same individual twice. This case normally only happens when the parent is very fit and chosen twice consecutively in a random process. The crossover in this case does not change the child but creates a copy of the parent, which is then however subjected to mutation.

To visualize this process we have created a flowchart in Figure 1.6. This process of the creation of a new generation and subsequent evaluation is repeated until either a whole generation or an individual has reached a certain threshold in fitness. GAs are relatively resistent to converging on local optima, but are not entirely immune to it. The convergence to a local optimum in GA terms is called premature convergence.

In summary, we have chosen GAs because they are known to be able to navigate in complex search spaces. In addition they are inherently parallelizable which makes it easy to speed up the optimization routine. For a more detailed overview we urge the reader again to consult Chapter **??**.

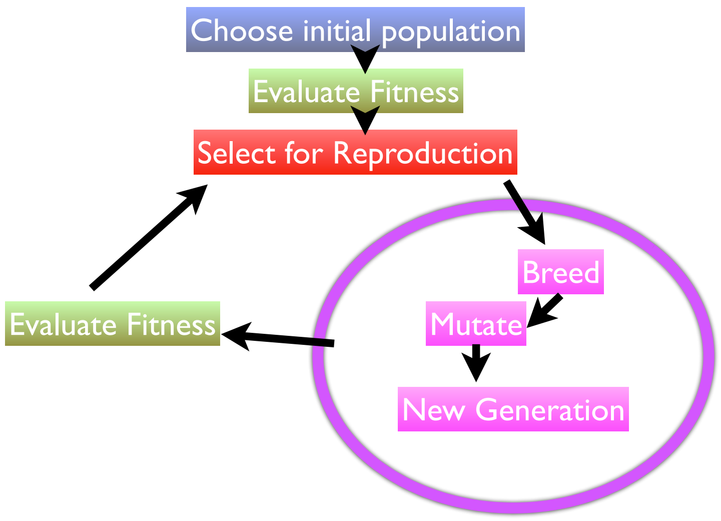

Figure 1.6 This gives an overview over the process of a GA

1.5. Genetic Algorithms fit Type Ia Supernovae: The Dalek Code

In the following section I will outline the current progress of the automation of fitting the SNe Ia spectra with the MLMC. The torrent of new spectra observed by a multitude of surveys requires the swift and self consistent analysis of many SNe Ia spectra in a short time. As described in Section 1.3 the search-space for fitting SNe Ia spectra is very complex and vast. It thus requires a smart and quick algorithm for analysis.

The parameters that need to be fit are luminosity, photospheric velocity (v_{ph}), carbon, magnesium, silicon, sulphur, calcium, chromium and nickel prior to decay (Ni_0) as well as stable iron (Fe_0). As in the manual fitting example the time since explosion as well as the luminosity distance (for scaling purposes) are given. We initially tried a Newton-Raphson approach with multiple phases. In the first phase the algorithm would adjust luminosity and normally came close to the optimum. In a second phase we tried to let the algorithm re-adjust luminosity, then photospheric velocity and last IGE s. This was modelled after the manual approach that is taken by Hachinger (2007) and (Hachinger, 2011). The process was time consuming and did not readily converge. We realized quickly the search space is far to complicated for such simple methods. In addition, we were limited to one processor with the Newton-Raphson-like method and could not utilize the large number of multi-processor machines that are currently available. GAs seem the perfect choice for this problem, as they are intrinsically parallel, are easy to implement and are relatively immune to local optima.

The first task was to wrap the MLMC code so that it is able to easily interface with any optimization routine and run in parallel in multiple instances. The initial version of this *launcher* code was able to run on multiple processors on one machine. Next we extended the *launcher* code to be able to distribute jobs on the network. As the institute has a heterogeneous computer structure (different processor types, different operating systems) we were forced to write a custom software to distribute it among the many nodes. An additional advantage to this cloud-like process structure is that we could use unused resources as well as avoid servers that are heavily demanded by other users. For the scheduling we used a simple first in, first out queue. Finally, the *launcher* code has a simple application program interface (API) that accepts arrays with parameter sets and returns the results in a simple array format. Invisibly to the user, it writes the parameter sets to the relevant disks in a format that the MLMC understands. Upon completion it reads the data files and cleans up any temporary files created by the MLMC. This abstraction of the MLMC allowed us to quickly explore the intricacies of the GAs (and in the future possibly other optimization algorithms).

As described in Section 1.4 the first step for any GA is to create an initial population (first generation). One can easily draw uniform randomly for luminosity and v_{ph} (within some bounds). However, this method does not work for the elemental abundances as the sum of the abundances needs to add up to one. As explained previously these abundances are abundances by mass with a pre-chosen W7 model density structure. A population that is distributed around the optimum value converges much quicker than a uniformly sampled one. Thus we have chosen to use the W7 model abundances as an input parameter as it seems to be a good starting point for many SN Ia. To calculate these abundances we need to know the photospheric velocity. Using data from Benetti et al. (2005) we have estimated an empirical relationship between the time since explosion and the photospheric velocity (see Figure 1.7). This will serve as a rough first guess.

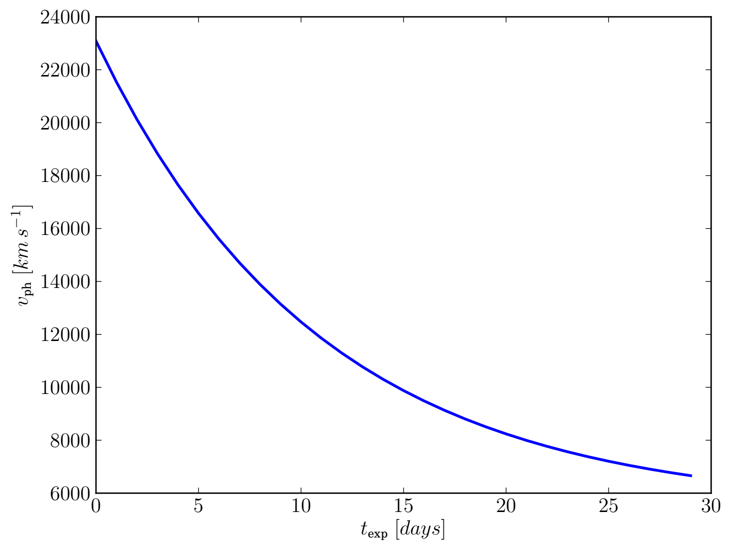

Figure 1.7 Estimated initial guess for photospheric velocity against days after explosion (Benetti et al., 2005).

Once we know a v_{ph}-estimate we can determine at what depth the photosphere is located in the W7 model. We use this point and integrate outwards to find our initial abundance fractions.

When creating our initial population, we draw randomly from a uniform distribution for a preselected luminosity range (for the moment determined manually) and a v_{ph} range 4000 km s^{-1}above and below the v_{ph} estimate. The random sampling of the elements poses some problems.There are a lower and an upper bound when it comes to the initial abundance ratios. We have a lower bound for each abundance (established as the smallest abundance an element can have by MLMC) at 10^{-7} and a upper bound which is determined by the requirement that all abundances need to add up to one. Knowing the minimum and maximum bound we then use a log-normal asymmetric random distribution around the actual W7 model value with the lower and upper bound being at three sigma. If we would draw the abundances every time in the same order it would mean that the last element to be drawn would always have a very low chance of obtaining a high abundance value. Thus we randomize the order of drawing the elemental abundances. This ensures that the initial population is sufficiently dispersed to avoid the GA's premature convergence on a local optimum, but close enough to the region of interesting parameter space, which avoids unnecessary delay in the convergence.

After creating an individual through the random drawing process, we also check abundance ratios allowed by explosive nucleosynthesis of a CO-WD outlined in Table 1.1. If the newly created individual does not conform with the abundance limits laid out in Table 1.1 it is discarded and a new one is drawn. Once the initial population (we chose 150 for our population size) is created it is distributed among the compute nodes by the *launcher* module of the Dalek code.

The resulting spectra are then subjected to a fitness function. The selection of the fitness function is one of the most crucial choices for any GA. The correct fitness function is still a field of active research in the Dalek code. As an initial approach we calculated the mean-square of the residuals remaining from subtracting the observed spectrum and the spectrum of the current individual.

This approach has two main issues. First for almost all observed spectra in the early phase the fitted spectrum has a large continuum excess beyond 6500 Å(see Figure 1.1). However, we still regard the line depth and line shape to be correct. As described previously this is a known issue of the MLMC code. This large difference in the infrared means that the Dalek code will try to optimize this large offset and pay less regard to a good fit in the rest of the spectrum. To alleviate this we have tried multiple approaches. At first we tried to de-weight the fit in the problematic region. This artificially introduced weighting factor introduces another parameter which might have to change it for the GA to succeed on different spectra. This would defeat the point of an automatic fit. As a second approach we tried to fit and subtract the continuum in the problematic region before creating our fitness figure.

Secondly there are different parameters, including the dilution factor, that help guide the fit which are neglected when just using the traditional mean-square approach. For example, the dilution factor is expected to be close to 0.5 (see description in Section 1.2). Hachinger (2007) and Hachinger (2011) have found that in all cases a good fit will have a value of the dilution factor between 0.4 and 0.7. We have incorporated this into the fitness calculation and de-weight the fitness of spectra with a value outside this range to a great extent. Due to both these reasons, we now include a traditional goodness of fit measure in the blue, the more complex fitness calculation described above and an inclusion of the dilution factor W in the final fitness.

We should mention that we have tried, in addition to calculating the root-mean-square of the spectral fit, a number of completely different methods. Most notably we tried to use neural networks to perform a goodness of fit analysis. We however abandoned this effort as the training set to calibrate neural networks requires a large number of well fitted spectra which are not available.

Once the fitness is calculated for each individual we use the method of fitness scaling to preempt premature convergence in early generations (caused by the so called 'super-individuals') as well as creating a steeper fitness gradient in later generations. We have decided to use a linear fitness scaling as described in Goldberg (1989, see page 76 of):

$$f' = af + b,$$

where $f\prime$ designates the scaled fitness and f the raw fitness. In all cases we want to make sure that the average of the scaled fitness f'_{avg} equals that of the average of the raw fitness f_{avg}. We will first try to find a linear relation so that the new maximum fitness f'_{max} is C_{mult} times the average fitness (f_{avg}). We choose $C_{\mathrm{mult}} = 2$ as suggested by Goldberg (1989). This operation will scale early 'super-individuals' down and the rest of the population up and preempts premature convergence. In the later phases of the GA, when the fittest individuals and the bulk of the population have similar fitness values, this operation would lead to negative fitness values for some individuals. In that case we find a scaling parameters that maps the least fit individual to a fitness value of 0, while still maintaining $f'_{\mathrm{avg}} = f_{\mathrm{avg}}$. Once we have scaled the fitnesses we move to the selection process.

In the current version of the Dalek code we employ *elitism* (10% of fittest individuals advance immediately to next generation). We have also experimented with other modifications to the mating population, but found them to be subpar. The mating pool we have chosen has only two slots. For now we select the individuals for the mating pool using the standard roulette wheel selection (RWS).

We use a crossover probability of 90%, this means that in 10% of the cases we do not perform crossover. This means that the child is a copy of the first parent. If crossover occurs, we perform a uniform crossover as we felt that we do not want the ordering of the parameters to matter. We have also experimented with arithmetic crossovers by taking the mean of the parents. This meant that often useful traits would entirely disappear and we switched back to uniform crossovers. We will experiment in the near future with different crossover techniques like single-point crossovers.

The new individual, created by crossover, is subjected to possible mutations before being placed into the new population. The Dalek code employs different mutation strategies for the different parameters. The abundances are just multiplied by a uniform random number (we have tried gaussian random numbers, but this led to premature convergence in many cases) with a relatively high chance of currently 7%. Luminosity and photospheric velocity are physically different from the element abundances and as such are treated differently in the mutation process. In both cases we add a uniform random number, in addition to having different mutation probabilities (this applies to all elements but oxygen). After the mutation step we see if the buffer element oxygen has a negative abundance. A negative abundance implies that the other elements add up to more than one. If oxygen has positive abundance the code goes on to check the abundance ratios. If the child passes both these tests it is placed in the new population, if not it is discarded. This process is repeated until the size of the new population (including the members of the old population that advanced through *elitism*) reaches 150.

This process of selection, recombination and mutation is repeated over many generations. We have experimented with up to a 1000 generations in one run. Our scheduler can achieve a throughput of one generation per minute (in optimal conditions where all CPUs are free). Our principal spectrum for testing the GA was SN 2002bo 10.4 days post explosion (see Figure 1.1). We have also tried different spectra with varying degree of success.

In the presented case we let the GA run for 188 generations. Figure 1.8 shows how the genetic algorithm over many generations improves the fitness of the individuals.

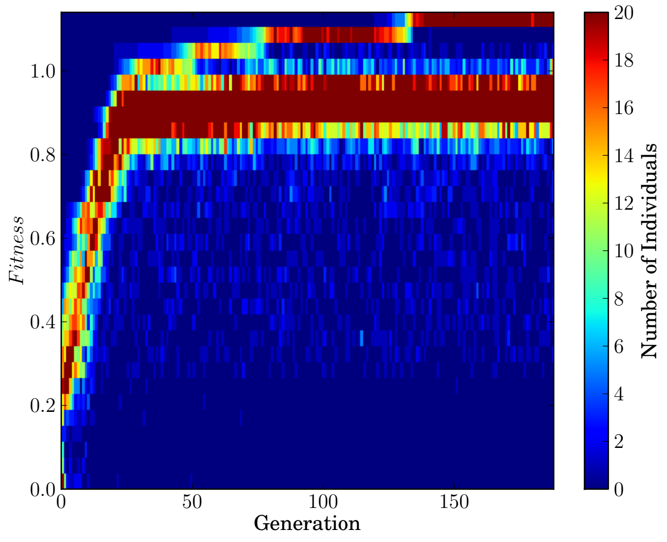

Figure 1.8 Evolution of fitness of in the generations. In later generations there is a clear divide between the members that have advanced through elitism and the bulk of the population. This suggests a too high mutationrate or the convergence of the algorithm.

The step-pattern is very common in genetic algorithms: One individual has a favorable mutation and it takes several generations for the other individuals to catch up. During the last generations one can clearly see that there is a gap between the bulk of the population and the top 10%. This gap is caused by the contrast of relatively high mutation in the main individuals against the mutation free advancement of the top 10% (elitism).

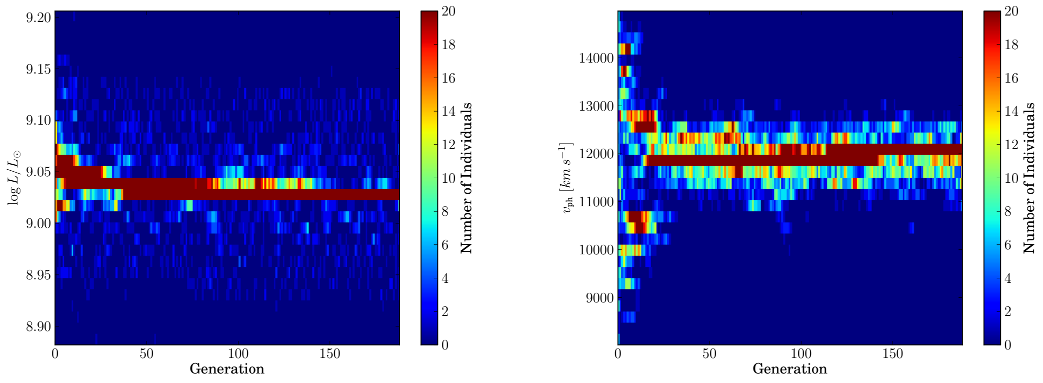

Figure 1.9 The evolution of both luminosity and photospheric velocity shows that even at late phases of the algorithm there are still new combination that are being trialled. The very quick convergence in both cases however is a bit worrisome and will be analysed in future experiments.

Figure 1.9 gives a good overview of how the genetic algorithm wanders through the parameter space. In our example the algorithm converges relatively fast (after roughly 30 generations). Even though there is a 'main'-population the algorithm still tries out different combinations. Currently we still have the problem that for certain initial random seeds the algorithm does not find the global optimum but converges prematurely on a local optimum. We believe the quick convergence seen in Figure 1.9 might be evidence for this. This again is an area of active research and two experts in evolutionary optimization joined our team recently to address this problem.

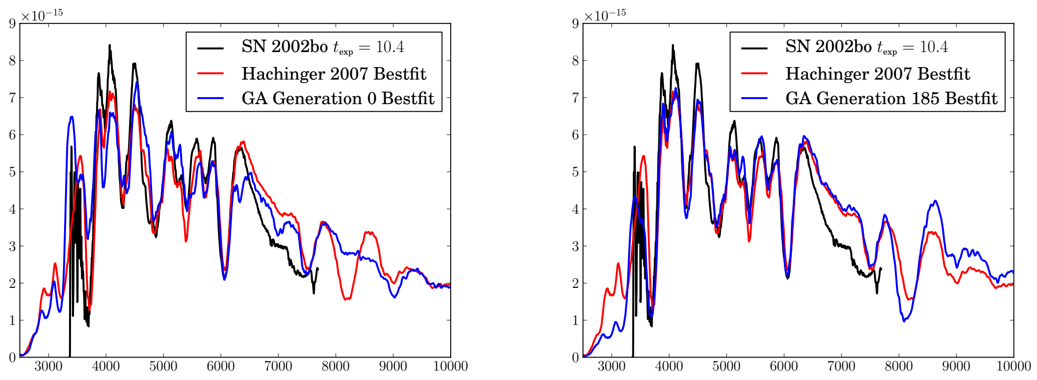

Figure 1.10 The best individual of the first generation (left) and the best individual after 185 generations (right). This initial result demonstrates that GAs are able to conquer this problem. A more stable convergence and thorough exploration of the parameter space is however necessary to use this technique on a set of supernova spectra.

Although there are still many outstanding problems we can show that the GAs are able to solve the SNe Ia fitting problem. In Figure **??** we show the best fit in the first generation as well as the best-fit in the last generation. As comparison we show the best fit obtained by Hachinger (2007). The GA has been successful in reproducing the main features of the SNe Ia-spectrum. The de-weighting in the infrared is working very well. Finally, we are currently trialing the Dalek code on different spectra with mixed results.

1.6. Conclusion

We have completed two major steps for the automation of SNe Ia fitting. First we have written an easy API that allows many synthetic spectra to be created simultaneously on an array of machines. Secondly we have demonstrated that GA are powerful tools which can find solutions efficiently in the vast parameter space of SNe Ia spectra. We do acknowledge that our current implementation does not reliably work on all spectra and there remains lots of fine-tuning work to be done. This tweaking is very common in GA implementations, but often requires experts in numerical optimization for speedy advancement. James Montgomery and Irene Moser are experts in evolutionary optimization (their research focuses on differential evolutionary algorithms) and are currently exploring with our help the search space.

Once the Dalek code is able to fit spectra reliably with the current code we plan to first use this on a large set of SNe Ia spectra and explore general abundance trends. Similar to Mazzali et al. (2007) we plan to unravel the stratification and mass of the ejecta. Depending on the explosion mechanism (deflagration or detonation) we will find different elemental abundances. Either explosion mechanism will likely hint at the mass of the white dwarf just prior to explosion. This will be an important indicator for the still unsolved progenitor question addressed in the other chapters of this thesis. Finally, we will plan to extend the Dalek code to use the stratified version of the MLMC, while much more complicated, will help us to improve the models of the layered structure of SN Ia.

On a different front we are also exploring ways to optimize the parameter space exploration speed. Even on current high-end machines the MLMC takes one minute per synthetic spectrum per CPU. One of the ideas is creating a parameter grid around all or some of our search area. We could then use fast and efficient interpolation techniques (one of these is described in Chapter **??**) and explore different settings for our genetic algorithm much more quickly.

Over the course of this project we have realized that the field of numerical optimization has made huge strides in the last decades. Most of the new algorithms are not well known in astronomy and are used even rarer. On the other hand experts in the field of numerical optimization are often in desperate need for 'real world' applications. They are more than willing to apply their knowledge to problems in our field. We strongly believe that there is an important ground for collaboration that would benefit both sides immensely.

BIBLIOGRAPHY

Benetti, S., et al. 2004, MNRAS, 348, 261 (ADS entry)

—. 2005, ApJ, 623, 1011 (ADS entry)

Branch, D., Fisher, A., & Nugent, P. 1993, AJ, 106, 2383 (ADS entry)

Chakraborty, U. K., & Janikow, C. Z. 2003, Information Sciences, 156, 253 , evolutionary Computation (Link)

Dorigo, M., & Gambardella, L. M. 1997, Ant colonies for the travelling salesman problem

Fisher, A. K. 2000, PhD thesis, THE UNIVERSITY OF OKLAHOMA (ADS entry)

Goldberg, D. E. 1989, Genetic Algorithms in Search, Optimization, and Machine Learning, 1st edn. (Addison-Wesley Professional) (Link)

—. 1990, Complex Systems, 5, 139

Hachinger, S. 2007, Master's thesis, TU München

—. 2011, PhD thesis, TU München

Hauschildt, P. H., & Baron, E. 1999, Journal of Computational and Applied Mathematics, 109, 41 (ADS entry)

Holland, J. H. 1962, J. ACM, 9, 297 (Link)

—. 1975, Adaptation in Natural and Artificial Systems (Ann Arbor, MI, USA: University of Michigan Press)

Iwamoto, K., Brachwitz, F., Nomoto, K., Kishimoto, N., Umeda, H., Hix, W. R., & Thielemann, F.-K. 1999, ApJS, 125, 439 (ADS entry)

Janikow, C. Z., & Michalewicz, Z. 1991, in Proc. of the 4th International Conference on Genetic Algorithms, ed. R. K. Belew & L. B. Booker (Morgan Kaufmann), 151–157

Kasen, D., Thomas, R. C., & Nugent, P. 2006, ApJ, 651, 366 (ADS entry)

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. 1983, Science, 220, 671 (Link)

Konak, A., Coit, D. W., & Smith, A. E. 2006, Reliability Engineering & System Safety, 91, 992 (Link)

Kromer, M., & Sim, S. A. 2009, MNRAS, 398, 1809 (ADS entry)

Mazzali, P. A. 2000, A&A, 363, 705 (ADS entry)

Mazzali, P. A., & Lucy, L. B. 1993, A&A, 279, 447 (ADS entry)

Mazzali, P. A., Röpke, F. K., Benetti, S., & Hillebrandt, W. 2007, Science, 315, 825 (ADS entry)

Michalewicz, Z. 1994, Genetic algorithms + data structures = evolution programs (2nd, extended ed.) (New York, NY, USA: Springer-Verlag New York, Inc.)

Mihalas, D. 1978, Stellar atmospheres /2nd edition/, ed. Hevelius, J. (ADS entry)

Nomoto, K., Thielemann, F.-K., & Yokoi, K. 1984, ApJ, 286, 644 (ADS entry)

Rudolph, G. 1994, Neural Networks, IEEE Transactions on, 5, 96 (Link)

Sobolev, V. V. 1960, Moving envelopes of stars, ed. Sobolev, V. V. (ADS entry)

Whitley, D. 1994, Statistics and Computing, 4, 65

Wright, A. H. 1991, in Foundations of Genetic Algorithms (Morgan Kaufmann), 205–218