

Spectral decomposition by genetic forward modelling

S.W. McIntosh¹, D.A. Diver¹, P.G. Judge², P. Charbonneau², J. Ireland³, and J.C. Brown¹

¹ Dept. of Physics and Astronomy, Kelvin Building, University of Glasgow, Glasgow, G12 8QQ, UK

² High Altitude Observatory, National Center for Atmospheric Research*, P.O. Box 3000, Boulder, CO. 80307-3000, U.S.A.

³ School of Mathematical and Computational Sciences, University of St. Andrews, St. Andrews, KY16 9SS, UK

Received January 15; accepted March 27, 1998

Abstract. We discuss the analysis of real and simulated line spectra using a genetic forward modelling technique. We show that this Genetic Algorithm (GA) based technique experiences none of the user bias or systematic problems that arise when faced with poorly sampled or noisy data. An important feature of this technique is the ease with which rigid a priori constraints can be applied to the data. These constraints make the GA decomposition much more accurate and stable, especially at the limit of instrumental resolution, than decomposition algorithms commonly in use.

Key words: Sun: UV radiation — methods: data analysis — techniques: spectroscopic — line: identification — profiles

1. Introduction

The recent launch of the SOLar and Heliospheric Observatory (SOHO) satellite, has renewed interest in the classification (Seely et al. 1997; Laming et al. 1997) and interpretation (Brekke et al. 1997; Judge et al. 1997) of high spectral resolution ultraviolet (UV) and extreme ultraviolet (EUV) emission spectra. The majority of these spectra come from the Solar Ultraviolet Measurement of Emitted Radiation (SUMER), and Coronal Diagnostic Spectrometer (CDS) instruments on-board SOHO (Wilhelm et al. 1995; Harrison et al. 1995).

A first step in the analysis of emission line spectra is to identify and measure properties of lines believed to be present. This is usually achieved by associating (subjectively) the observed spectral profiles with ionic and atomic transitions of “known” laboratory wavelengths.

From these possibly biased decompositions, physical models of the underlying plasma are sought. In an effort to obtain the best possible scientific results from their spectra, the CDS and SUMER teams have set about ways to produce the most “reliable” decomposition; see Brynildsen (1994) for more details.

Standard spectral decomposition techniques unfortunately prove to be unstable when presented with data of low signal to noise ratio, or data that is poorly sampled. In particular these instabilities cause subtle differences in the decomposition of each spectrum and can lead to significantly different physical interpretations. This has prompted us to search for a method that can provide spectroscopists with reliable decompositions of observed spectra that are as free as possible from subjective bias.

We use a heuristic approach to decomposition. We use a Genetic Algorithm (GA) to fit model line profiles, which for our purpose we chose to have Gaussian form, to provide a simple parameterisation of the spectrum under analysis. This approach exploits the stability and optimization capabilities of natural selection (Darwin 1859). Sections 2.1 and 2.2 describe the basic GA formalism, and an introduction to our Gaussian fitting GA, hereafter Ga-GA.

The GA technique is applied under ideal conditions (to “simple” noiseless test spectra) in Sect. 3.1. This first test also helps to highlight how well genetic operators are suited to this task. Section 3.2 gives a much more stringent test of the how a GA performs when fitting spectra containing unstructured random noise. Here, the GA’s stability in the presence of random Gaussian noise is compared to that of standard profile fitting and optimization algorithms. We show that these standard algorithms are blighted by possible user bias which is *not* present with the GA technique. To aid further comparison of our GA technique to standard analysis algorithms we have constructed model spectra with realistic noise and continuum/background levels. The results are discussed in Sect. 3.3.

Send offprint requests to: S. McIntosh

* NCAR is sponsored by the National Science Foundation.
Correspondence to: scott@astro.gla.ac.uk

The ability of the GA approach is given a final test in Sect. 4 on quiet Sun SUMER spectra. There we compare our results with those obtained from an analytical decomposition performed by Judge et al. (1997). We note that their technique used additional information not available to the GA.

Although much emphasis must be placed on the fact a GA requires minimal user input, in certain circumstances user input can prove useful, such as cases where relative wavelengths and intensities are well known from atomic physics. Such additional constraints can (almost trivially) be “hard-wired” into the algorithm. Section 4.1 highlights the possibilities of applying rigid a priori constraints to the observed spectrum.

2. Motivation and method

Prior to the launch of SOHO, a study was undertaken (Brynildsen 1994) to identify the “best” profile fitting package for the CDS and SUMER instruments. The study compared various algorithms for fitting Gaussian profiles, or combinations thereof.

The common denominator linking all of the profile fitting algorithms studied by Brynildsen (CURVEFIT - from the Interactive Data Language (IDL) userlib, and AMOEBA - A “downhill” SIMPLEX algorithm from Press et al. 1992, and others) is the need for user input regarding starting points for *each* parameter in the search. This potential source of user bias, and the reduced quality (in terms of fit to the data) of the parameterisation form the principal motivation for this paper, and indeed we show that they are not present using a GA technique beyond the absolute minimum requirement of supplying a “line list” of lines to be identified.

Using a GA for this profile fitting problem can have many advantages not available to the user of predictive line fitting algorithms. Considering one of the many advantages noted in Charbonneau (1995), a GA is not destabilised by noise in the data; it will merely attempt to achieve its goal, locating the “best” profile. The GA will attain this goal, the introduction of data noise will merely affect the convergence time of the algorithm.

We present a “simple” GA, called Ga-GA, which we show to be stable against reasonable noise levels and to have no source of possible user bias. The following sections discuss its performance in detail.

2.1. Overview of a simple Genetic Algorithm

Genetic Algorithms are inspired by the mechanism of natural selection and basic genetic operators, occurring naturally in biological systems, see Holland (1962). Consider a typical numerical optimization task, where a parametric model is to be fit to data in a manner that maximises the

closeness of fit, or *fitness* (as measured, for example, by a χ^2 statistical estimator). A genetic algorithm is an iterative scheme that operates on a *population* of trial solutions to the problem in the following way:

1. Construct an initial population using *random* values for the model parameters, and evaluate their fitness.
2. Select a subset of the fitter individuals, and breed them to produce a new population.
3. Evaluate the fitness of each individual in the new population.
4. Replace the old population with the new one.
5. Check whether the fitness has reached some predefined tolerance, or the number of iterations (or *generations*) has reached its maximum; if not return to step 2.

GAs carry out a form of forward modelling, by performing a heuristic search of the problem’s parameter space. What distinguishes a GA from other forward modelling methods (such as Monte Carlo simulation) is primarily the way in which new trial solutions are constructed from the current population of trial solutions (cf. step 2 above).

At the most basic level a GA can be viewed as a processor of a set of strings, each encoding a particular version of the model being optimised. A subset of the fitter individuals of the current population are selected and paired, and the defining strings of each such pair are subjected to the action of two genetic operators: *cross-over* and *mutation*. The cross-over operation involves dissection of the two parent strings at a randomly chosen point along the string, followed by the interchange of the dissected components. In this way two new strings are produced from two parent strings (see Fig. 1). The second operator, *mutation*, involves the replacement of a few randomly selected digit in the two strings produced by the cross-over operation with randomly generated digit values. Its primary purpose is to maintain a suitable level of variation in the population, which is essential for selection to operate. The combination of stochastic genetic operators with fitness-based selection yields a powerful search algorithm that can move away from secondary extrema and locate the *global* extremum in parameter space (see, e.g., Goldberg 1989; Davis 1991; Charbonneau 1995)

In this paper we are using a GA version which implements a scheme involving a *variable* mutation rate, i.e. as the population becomes more degenerate (little variation) the probability of a mutation taking place is correspondingly increased, and makes use of *elitism*, the best individual in the old generation is *not* replaced unless there is a fitter one in the new generation. The selection of individuals in the breeding operator is carried out using a *roulette-wheel* algorithm (see Davis 1991, Ch. 1), meaning that individuals with higher fitness are associated with sectors of correspondingly large angle on the roulette wheel. This roulette wheel, when “spun”, ensures that although all

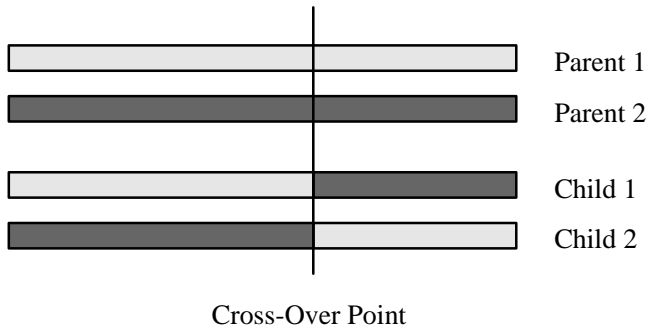


Fig. 1. A pictorial explanation of the main GA breeding operator, *cross-over*

individuals are capable of breeding, the fitter individuals have a slightly higher probability of being selected.

In many ways our GA resembles that of Charbonneau (1995), but it also contains some features of the GA presented in Diver & Ireland (1997). Indeed, in the cases presented in Sect. 4 we have employed a variation on the algorithm *PIKAIA* presented in Charbonneau (1995) to maximise speed and accuracy.

2.2. Fitness evaluation

Isolation of particular features (e.g. line width and absolute intensity) in an emission line spectrum made up of N lines is a procedure used by many standard fitting algorithms, with many using line identification as their primary “search” (cf. the user input given to the algorithms mentioned above). On acquiring the line position they sequentially alter the amplitude or the $\frac{1}{e}$ width of the Gaussian profile(s) to achieve the “best” fit to the target. However, since the observed emission line spectra can and do, contain a large number of profiles, it is possible to adopt a method which solves for all lines simultaneously (see e.g., Diver 1995; Diver & Ireland 1997).

When Ga-GA “recognises” spectral features, i.e. one of the Gaussian describing parameters or an entire profile, the corresponding final solution will be a better representation of the target and will result in that string of parameters being given a higher fitness. Since Ga-GA uses the mechanics of natural selection, a genotype with a higher fitness value will be prevalent in the current and future generations until replaced by a “fitter” individual.

Ga-GA uses parameter strings of length $3 \times N$, where N is the estimated number of Gaussian profiles in the line spectrum to be analysed, and three because it requires three parameters to describe a general Gaussian profile. These parameters are absolute position in wavelength, at channel (X), amplitude (A), and the Gaussian’s $\frac{1}{e}$ value (W) and are encoded as a string in the following order:

$$[X_1, A_1, W_1, \dots, X_N, A_N, W_N].$$

A string of the form above defines a sequence of N Gaussian profiles that defines a synthetic spectrum, this

computed profile is an individual’s phenotype. It is this phenotype profile that is retained for comparison to the observed spectrum. Phenotype profiles are calculated using the standard pointwise Gaussian formula, i.e. for a particular channel x , usually associated with wavelength, in Gaussian i (G_i):

$$G_i(x) = A_i \exp\left(\frac{-(x - X_i)^2}{W_i^2}\right). \quad (1)$$

The N Gaussian profiles derived from a particular genotype string are summed to form the “unique” phenotypic profile for genotype j , $P(\underline{x})_j$ (with \underline{x} meaning for all channels x). $P(\underline{x})_j$ is given by:

$$P(\underline{x})_j = \sum_{i=1}^N G_i(x) \quad \forall x. \quad (2)$$

Only once $P(\underline{x})_j$ has been computed do we calculate an *error* measure between it and the target. The error measure of a particular genotype ($E(\underline{x})_j$) depends on several factors; the square pointwise difference of the target and the corresponding phenotype ($C(\underline{x})$, and $P(\underline{x})_j$), the number of parameters in the calculation ($3 \times N$), the number of points summed over (N_{data}) and an estimate of the noise level in the data ($\sigma_{\text{data}}(\underline{x})$). Thus, $E(\underline{x})_j$ (effectively a normalised χ^2 measure) is given by:

$$E(\underline{x})_j = \frac{1}{(N_{\text{data}} - 3N)} \sum_x \left(\frac{(C(\underline{x}) - P(\underline{x})_j)^2}{\sigma_{\text{data}}(\underline{x})} \right) \quad (3)$$

with $E(\underline{x})_j \sim 1$ indicating a “good” fit.

This measure is used to evaluate the fitness of each genotype. It is the fitness value that is used to rank all the genotypes in a particular population into ascending order and to “weight” the roulette wheel of Sect. 2.1.

3. Results

This section details the results of Ga-GA applied to simulated target data sets which have a known level of noise added. Section 3.1 discusses the performance of Ga-GA in the absence of data noise (except for very small numerical rounding errors). Sections 3.2 and 3.3 provide ideal circumstances to test the performance of Ga-GA, against that of the two standard algorithms mentioned earlier; CURVEFIT and AMOEBA, for data with a realistic noise level and with a noisy background present (Sects. 3.2 and 3.3 respectively). Section 3.3 will also show the ease with which additional spectral features may be incorporated into the analysis.

3.1. Application to noiseless target spectra

We use Ga-GA to analyse three noiseless targets, i.e. we replace $\sigma_{\text{data}}(\underline{x})$ by 1 in Eq. (3), each corresponding to a different Gaussian configuration. The three test

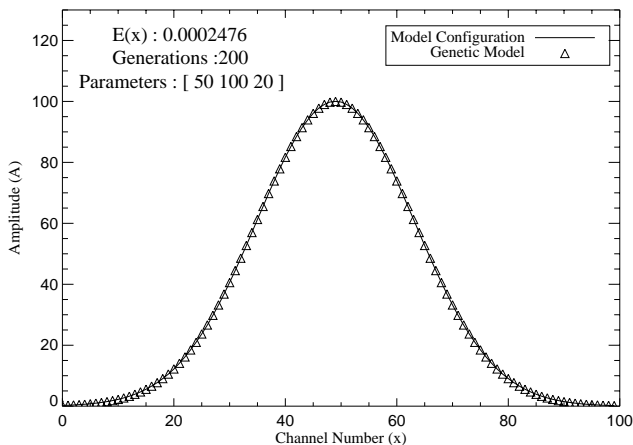


Fig. 2. Test run for Ga-GA, taken from the ensemble of ten runs, for the *noiseless* single Gaussian target (solid line) of Case 1 and the profile modelled by Ga-GA (Δ)

targets are: 1) A single “wide” Gaussian with the target genotype given by three parameters, $[XAW] = [50\ 100\ 20]$. 2) Two “joined” Gaussians corresponding to the six parameter genotype $[40\ 100\ 20\ 80\ 90\ 15]$, and 3) a more complex five Gaussian configuration with the fifteen parameter target genotype given by $[10\ 30\ 5\ 22\ 60\ 1\ 26\ 40\ 3\ 43\ 70\ 5\ 55\ 60\ 5]$.

Each case was analysed ten times (to allow performance statistics to be compiled), each run with a different initial population, for a fixed number of generations. It is also possible to configure Ga-GA to run until it achieves a fixed $E(\underline{x})$ although for certain types of analysis this method is unfavourable (Charbonneau & Knapp 1996). The number of generations used in each case is different however, and varies with the increase in complexity of the target solution. Therefore target 3 typically requires a 1200 generation run, which is considerably more than the 200 and 500 generation runs required for targets 1 and 2 respectively.

The returned parametrisation of each target is given in Table 1. The subscript T quantities (e.g. X_T) are the target parameters and the subscript G quantities (e.g. X_G) are the corresponding mean values returned by Ga-GA after multiple fixed generation runs. It is clear from the results presented in Table 1 that Ga-GA obtains a *very* good representation of each target (within the errors). The errors in the parameters are global error estimates and are calculated in a Monte Carlo fashion, i.e. we perform multiple runs of Ga-GA each with a different initial population, this is achieved by initializing the random number generator with a different seed (Charbonneau & Knapp 1996). This Monte Carlo approach “forces” Ga-GA to search the parameter space from a different starting point each time. This will also allow the calculation of “mean” values for each of the parameters.

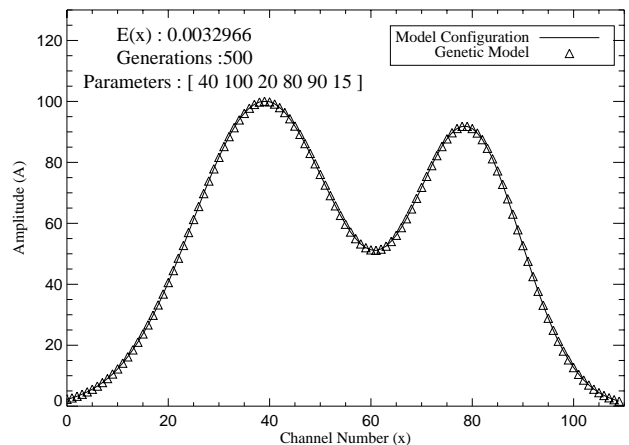


Fig. 3. Test run for Ga-GA, taken from the ensemble of ten runs, for the *noiseless* double Gaussian target (solid line) of Case 2 and the profile modelled by Ga-GA (Δ)

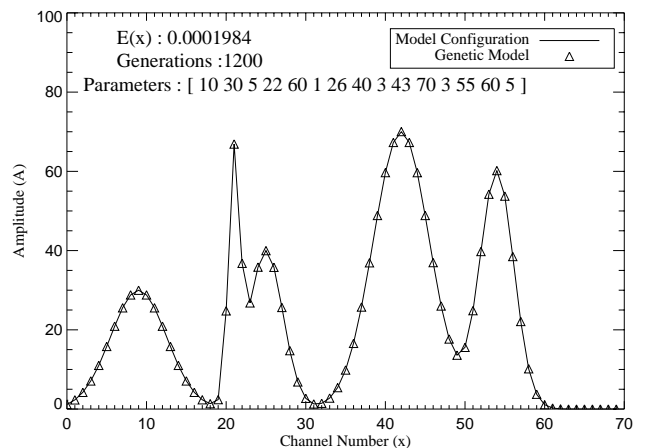


Fig. 4. Test run for Ga-GA, taken from the ensemble of ten runs, for the *noiseless* five Gaussian target (solid line) of Case 3 and the profile modelled by Ga-GA (Δ)

Figure 2 shows a plot of target 1 (solid line) and the profile derived from the “fittest” genotype (Δ) after only 200 generations with the $E(\underline{x}) = 2.476 \cdot 10^{-4}$. Similarly, Fig. 3 shows the profile constructed from the fittest genotype, $E(\underline{x}) = 3.296 \cdot 10^{-3}$, for the double Gaussian configuration of target 2. Figure 4 demonstrates Ga-GA’s handling of the more complex case 3, resulting in $E(\underline{x}) = 1.984 \cdot 10^{-4}$ of the fittest genotype after 1200 generations. For these test cases final values of $E(\underline{x})$, if we doubled the number of generations, will be limited by numerical precision and would possibly attain no better values than those given and it must be emphasised that these results are for *one* particular run of Ga-GA from the ensemble of 10 runs.

We show, in Fig. 5, the decrease in $E(\underline{x})$ with generation number for the full ensemble of runs (indicating the mean $E(\underline{x})$ (solid line), extrema (dashed line) and median (dotted line) for each generation step) for each of the test

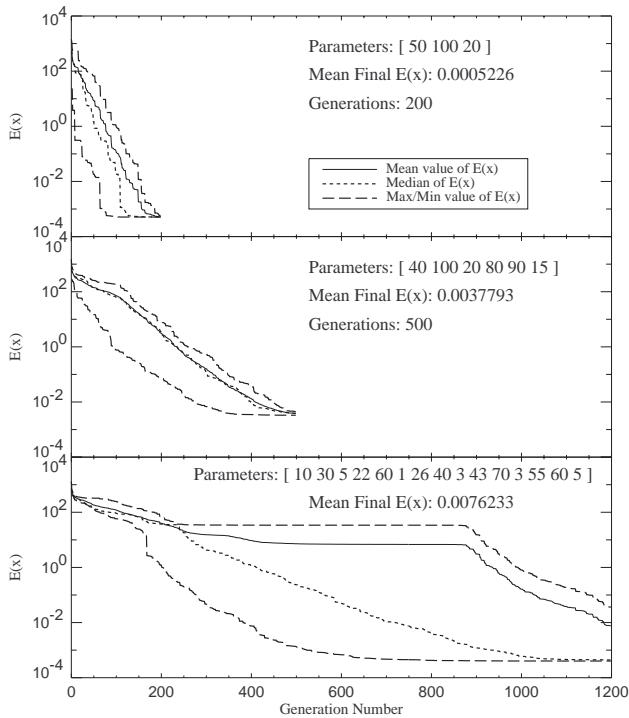


Fig. 5. Convergence of $E(\underline{x})$ against generation number for each of the three cases in Sect. 3.1. Top panel: Case 1 (single Gaussian), Middle panel: Case 2 (two Gaussians) and Bottom panel: Case 3 (five Gaussians). For each generation step the mean $E(\underline{x})$ (solid line), median (dotted line) and extrema (dashed lines), for the ten run ensemble, are indicated. It is clear that, when a relatively “poor” parameterisation is present, the difference between the median and mean of $E(\underline{x})$ is demonstrably effected, this effect is evident in the top and bottom panels

cases above. These plots demonstrate the power of Genetic Algorithms as optimization tools. The steplike structure is clearly visible in all three plots, although to a much greater extent in the uppermost plot. Such steps occur when Ga-GA suddenly obtains a new “fitter” value for one (or more) parameter(s), the long flat “plateaus” are points where the current “best” in the population hasn’t changed or when the population is largely degenerate, i.e. all the individuals have very similar genotypes. These mutation jumps will occur because the mutation rate has been allowed to increase, and will thus introduce new genetic material at a higher frequency.

Figure 5 also justifies our earlier claim that more complex targets (more parameters) require a greater number generations in the run. As with any optimization method the plots show how the gradient of $E(\underline{x})$ lessens with the increase in the number of parameters in the genotype, the increase in the number of generations required for a GA to evolve an acceptable solution increases with the dimension, D , of the search space; typically it does so in a manner that is highly problem dependent, but often ends

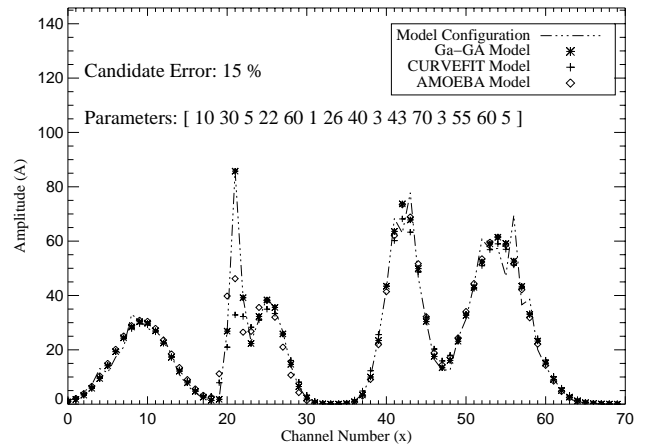


Fig. 6. Performance comparison plot between Ga-GA, AMOEBA and CURVEFIT. They are compared using the target of Sect. 3.2 with 15% added random noise. See also Table 2

up as being a low (order unity) power of N . So such convergence plots provide evidence to suggest that we have not yet evolved a “perfect” match for the target. This may be estimated by looking at the gradient of the plot at the end of its evolutionary run. The center and bottom plots in Fig. 5 show that the evolutionary process may not be finished.

3.2. Application to a “noisy” target spectrum

Reliable analysis of a “noisy” target must be the benchmark for any spectral decomposition technique. We therefore compare the performance of Ga-GA to that of the AMOEBA and CURVEFIT algorithms in decomposing a “noisy” five Gaussian target, again with Ga-GA results the mean of ten runs. The target is generated by the same fifteen parameter genotype as case 3 of Sect. 3.1 ([10 30 5 22 60 1 26 40 3 43 70 3 55 60 5]) to which we now add 15% “random” noise. The noise is set to be normally distributed about the data with an rms amplitude of 15%, so $\sigma_{\text{data}}(\underline{x}) = 0.15 C(\underline{x})$ in Eq. (3).

The results of the calculations for each algorithm¹ are shown in Table 2 where Ga-GA achieves the lowest $E(\underline{x})$ (1.889), by a factor of six from CURVEFIT (12.961) and by a factor of about ten from AMOEBA (18.626). It must be noted that all produce “good” parameterisations of the spectrum given the severe noise present, but bear in mind that the latter two algorithms are practically given the target parameters as a startpoint, and are hence heavily influenced by the user. This is definitely not the case with Ga-GA. CURVEFIT and AMOEBA also exhibit another behavioural pattern not observed with Ga-GA; they

¹ It should be noted that CURVEFIT and AMOEBA were initialised with a guess of each parameter that is within ± 2 the target parameter value.

Table 1. Results for cases 1), 2) and 3) described above. Subscript T quantities indicate target parameters, and subscript G quantities are the mean after multiple evolutionary runs. Similarly, the values of $\langle E(\underline{x}) \rangle$ are the final mean values of $E(\underline{x})$. The errors for each parameter are calculated as the means of the ten run ensemble

X_T	A_T	W_T	$X_G \pm \delta X_G$	$A_G \pm \delta A_G$	$W_G \pm \delta W_G$
	Case 1.	$\langle E(\underline{x}) \rangle$	$5.226 \cdot 10^{-4}$	200 gens.	
50.00	100.0	20.00	50.000 ± 0.000	100.002 ± 0.003	20.002 ± 0.001
	Case 2.	$\langle E(\underline{x}) \rangle$	$3.779 \cdot 10^{-3}$	500 gens.	
40.00	100.0	20.00	40.002 ± 0.002	100.007 ± 0.007	20.004 ± 0.003
80.00	90.00	15.00	79.997 ± 0.002	89.998 ± 0.003	14.999 ± 0.002
	Case 3.	$\langle E(\underline{x}) \rangle$	$7.623 \cdot 10^{-4}$	1200 gens.	
10.00	30.00	5.000	9.998 ± 0.001	30.003 ± 0.019	4.997 ± 0.004
22.00	60.00	1.000	21.997 ± 0.001	59.661 ± 0.181	0.995 ± 0.002
26.00	40.00	3.000	25.983 ± 0.007	39.867 ± 0.068	3.002 ± 0.006
43.00	70.00	3.000	43.000 ± 0.000	69.951 ± 0.028	3.001 ± 0.001
55.00	60.00	5.000	54.999 ± 0.001	59.964 ± 0.014	5.003 ± 0.001

will occasionally become “stuck” at points in the solution space where hope of convergence to the target is lost². This does not happen in every run, but indicates to the user that a single run using either method is not enough to guarantee a reliable parameterisation.

Figure 6 shows the results of Ga-GA (*), CURVEFIT (+) and AMOEBA (\diamond) operating on the fifteen parameter, five Gaussian target. The profile shown for Ga-GA, as in Sect. 3.1, is the “fittest” phenotype from the ten different runs. It is clear from the results in Table 2, and the plots in Fig. 6 that the sharp features of Gaussian two (at a possible limit of resolution) present CURVEFIT and AMOEBA with a very awkward test. Indeed, by inspection of the errors quoted in Table 2 it is possible to see the feature(s) that Ga-GA finds most awkward to “identify”, these are the amplitudes A_2 , A_3 and A_4 .

3.3. Application to a target with a background level

We now consider the case where the target has a considerable background level. A GA approach makes inclusion of such a background, or continuum, extremely simple. To show this, consider a parameterisation of the background by addition of a quadratic of order n , an example for $n = 2$ is given in Eq. (4). As an example, consider a new three Gaussian configuration [10 90 6 50 70 3 80 40 4] with 5% noise ($\sigma_{\text{data}}(\underline{x}) = 0.05 C(\underline{x})$) and background; the alteration to the fitness evaluation routine is minimal. We add the quadratic form to the standard phenotype calculation of Eq. (1), which then becomes:

$$P(\underline{x})_j = a + bx + cx^2 + \sum_{i=1}^N G_i(x) \quad (4)$$

where a , b , and c are taken from the adapted genotype by adding [$a b c$] to the Gaussian description parameters.

² The interested reader is directed to Charbonneau & Knapp (1996) for a discussion of this effect.

Table 2. Details of the target parameters (P_T), genetically modelled solution returned by Ga-GA and the deterministic routines for the fifteen parameter configuration with 15% normally distributed random noise. Ga-GA results and CPU times (T_{CPU}) are the mean of an ensemble of ten runs. The CPU times are normalised to the CPU time of a CURVEFIT run

P	P_T	AMOEBA	CURVEFIT	Ga-GA
X_1	10.00	10.340	9.317	10.305 ± 0.001
A_1	30.00	31.002	29.700	30.433 ± 0.011
W_1	5.000	5.101	5.062	4.908 ± 0.003
X_2	22.00	21.552	21.092	22.024 ± 0.001
A_2	60.00	45.021	27.985	81.160 ± 0.073
W_2	1.000	1.253	1.744	0.947 ± 0.001
X_3	26.00	25.790	25.305	26.187 ± 0.009
A_3	40.00	38.408	35.121	38.506 ± 0.031
W_3	3.000	2.843	3.051	2.856 ± 0.006
X_4	43.00	43.210	42.100	43.122 ± 0.001
A_4	70.00	73.502	67.914	73.611 ± 0.017
W_4	3.000	2.915	3.138	2.919 ± 0.001
X_5	55.00	54.887	54.016	55.018 ± 0.001
A_5	60.00	61.449	59.015	61.433 ± 0.001
W_5	5.000	5.055	5.265	5.050 ± 0.001
T_{CPU}		114.125	1.000	109.312
$E(\underline{x})$		18.626	12.961	1.889

To generate the target the background parameters are assigned the values $a = 30.0$, $b = 0.5$ and $c = 0.002$.

A plot of the target solution (broken line) and the best phenotype (*) is shown in Fig. 7. The figure also shows the profile returned by CURVEFIT (+) and that returned by AMOEBA (\diamond). Ga-GA’s estimate of the background parameters are $a = 29.243$, $b = 0.554$ and $c = 0.002$ (with respective errors given below). Ga-GA results were returned after 1000 generations and the mean final $E(\underline{x})$ was 0.8664, with CURVEFIT giving a statistically equivalent fit (0.8600) and AMOEBA by a factor of two (2.000).

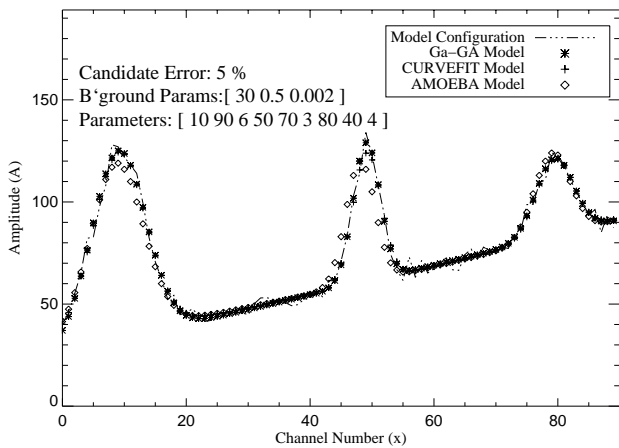


Fig. 7. Plot of the 3 Gaussian configuration [10 90 6 50 70 3 80 40 4] and the background parameters, $a = 30.0$, $b = 0.5$, and $c = 0.002$ with a 5% random noise level. See also Table 3

Table 3. Results from Sect. 3.3 for a target ($P(T)$) with fixed background level and 5% normally distributed random noise. Again, Ga-GA results and CPU times (T_{CPU}) are the mean of an ensemble of ten runs. CPU times are normalised to that of a CURVEFIT run

P	P_T	AMOEBA	CURVEFIT	Ga-GA
X_1	10.00	9.810	9.172	10.131 ± 0.017
A_1	90.00	82.404	88.062	90.926 ± 0.529
W_1	6.000	6.020	5.980	6.045 ± 0.056
X_2	50.00	49.100	49.160	50.101 ± 0.001
A_2	70.00	60.001	63.526	68.474 ± 0.121
W_2	3.000	3.312	3.238	3.059 ± 0.009
X_3	80.00	80.103	79.469	80.429 ± 0.001
A_3	40.00	41.261	37.544	38.340 ± 0.077
W_3	4.000	4.121	4.303	4.357 ± 0.015
a	30.00	31.180	31.740	29.243 ± 0.964
b	0.500	0.501	0.486	0.554 ± 0.037
c	0.002	0.002	0.002	0.002 ± 0.000
T_{CPU}		80.134	1.000	75.321
$E(\underline{x})$		2.000	0.8600	0.8664

The full results of the parameterisation for all three algorithms are given in Table 3.

4. Analysis of a quiet Sun SUMER spectrum

To test Ga-GA on real data we chose to analyse a spectral region in the SUMER wavelength range that is known to suffer from blending problems, both between spectra of different optical orders as well as just wavelength coincidences. Those problems resulting from blends between lines that happen to overlap in the first and second grating orders can be decomposed experimentally, and thus serve as a limited check on the the GA approach.

The dataset analyzed here was obtained on October 26th 1996, with the 1×300 arcsecond slit crossing the

north polar limb, using SUMER’s B detector. Data were acquired in the 1400 Å spectral region, containing strong lines of Si IV, O IV, and O III (in second order), as well as other weaker lines.

The observing sequence was designed to obtain data between 1399 and 1408 Å (and in the second order spectrum with wavelengths at half of this range) on both the bare and KBr coated part of the detector, sequentially. The exposure time on the KBr part was 180 seconds, and 360 seconds on the bare part. The bare and KBr regions of the detector have very different sensitivities to first and second order spectra. Assuming that the spectra did not change significantly between the bare and KBr exposures, the different count rates acquired on the two regions allow one to decompose the spectrum analytically into first and second order components, I_1 and I_2 through the following equations

$$Cts(\text{KBr}) = k_1 I_1 + k_2 I_2 \quad (5)$$

$$Cts(\text{bare}) = b_1 I_1 + b_2 I_2 \quad (6)$$

where $Cts(\text{KBr})$ and $Cts(\text{bare})$ refer to the count rates per pixel per second on the KBr and bare parts of the detector, I_1 , I_2 are intensities of the first and second order spectra, and k_1 , k_2 , b_1 , b_2 , are (known) instrumental sensitivities defined through these equations. Figure 8, top panel, shows $Cts(\text{KBr})$ and its components, $k_1 I_1$ and $k_2 I_2$. Values for I_1 and I_2 were obtained using measurements of $Cts(\text{KBr})$, $Cts(\text{bare})$ and instrumental sensitivities discussed by Judge et al. (1997). Figure 8 also shows $Cts(\text{bare})$ and its components, in the bottom panel. In each case the count rates are averaged over 300 spatial pixels, including the solar limb, and time during the exposures.

Shown in the top panel of Fig. 9 is a decomposition performed using Ga-GA based *only* upon the $Cts(\text{KBr})$ spectrum shown in the upper panel of Fig. 8. This is simply a “blind” fit, using *no* prior information about the spectrum, except that we expect between 16 and 20 Gaussians to be present with on constant background. Such “blind” fits show that we can obtain a reliable decomposition of the entire spectrum. An example where a “blind” run is significantly better than one where a priori knowledge is used to aid in the decomposition is given below (see Table 4).

4.1. Using additional knowledge

Usually, extra information about the spectrum is known, and it may be needed for some cases. This information can be “hard-wired” into Ga-GA easily. For example, we could demand that the spectral decomposition must not contain spectral detail narrower than the instrumental width (σ_{inst}). Or, we could specify that relative positions (or intensities) of lines from the same ion, known to great accuracy from laboratory measurement, be fixed to certain values. Such constraints can be incorporated into the GA

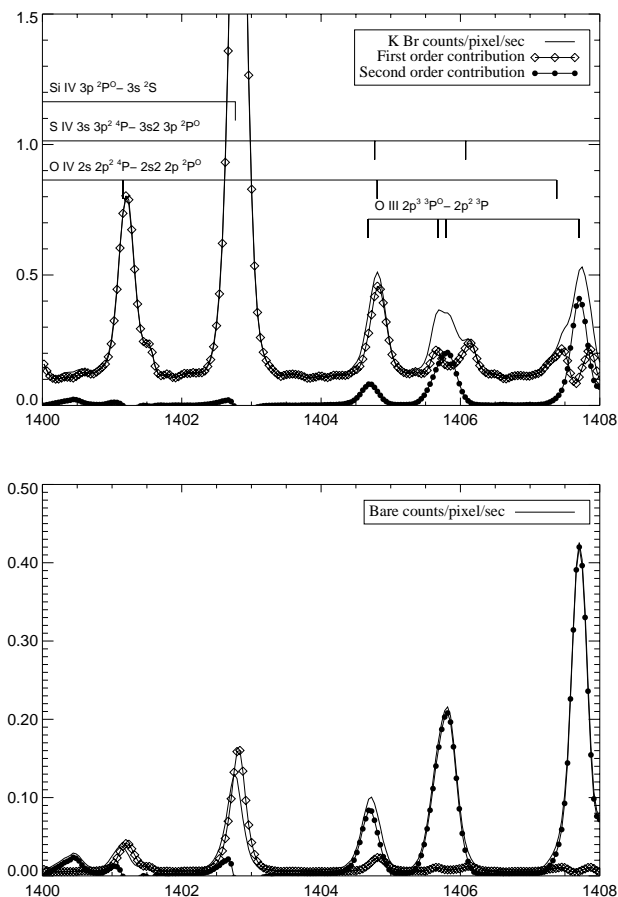


Fig. 8. The 1400 Å region of the solar spectrum as measured using the SUMER instrument (see text for details). The top panel shows the average spectrum, in counts/pixel/second, recorded on the KBr region of the detector. Positions of known strong lines are marked- the positions of lines of O III are marked assuming that they are formed in the second order. The bottom panel shows the same thing, but recorded on the bare part of the detector. The lines plotted with symbols show the spectral decomposition into first and second order lines using the known sensitivities from SUMER

through a simple modification of the fitness evaluation, Eq. (3). For such an example we might use:

$$E(\underline{x}) = \chi^2 + C_i H^2(W_i, \sigma_{\text{inst}}) + D_{ij} ((X_i - X_j) - (X_i^{\text{lab}} - X_j^{\text{lab}}))^2 + \dots \quad (7)$$

where we introduce the additional constants C_i and D_{ij} to control the “trade-off” between χ^2 and the newly incorporated information, and where $H(W_i, \sigma_{\text{inst}})$ will weight the optimization against features narrower than σ_{inst} . A future version of Ga-GA may take advantage of this additional information to act as desktop on-line plasma analysis package. Recall however, that the number of parameters in the calculation effects the rate of convergence (Sect. 3.1 and Sect. 3.2).

The lower panel of Fig. 9 shows the results of a Ga-GA decomposition where we have included a line list of all

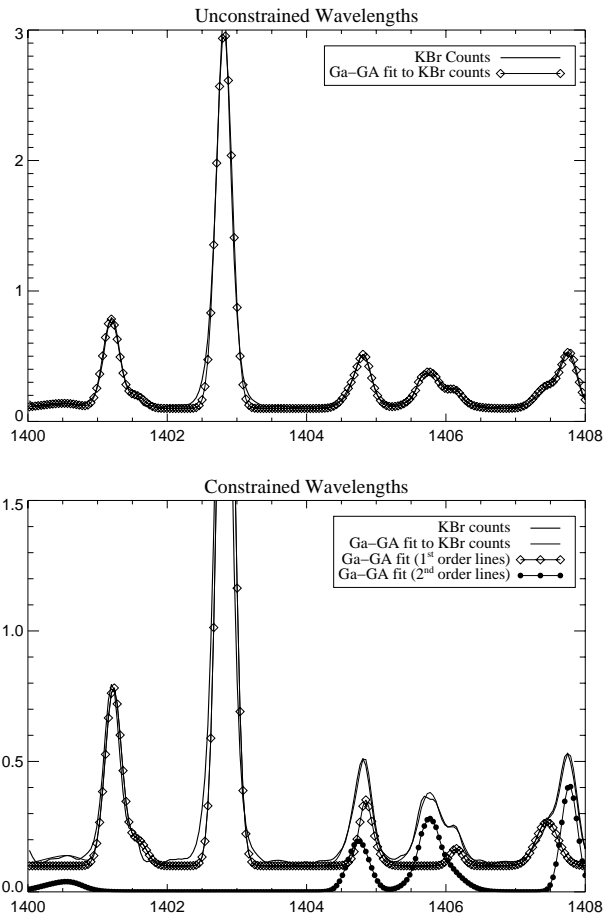


Fig. 9. Comparison between Ga-GA decomposition and the analytic decomposition of the SUMER spectrum in Fig. 8. The top panel shows the decomposition from the Ga-GA algorithm using only the KBr data from the top panel of Fig. 8. The bottom panel shows the decomposition from a *single* run of Ga-GA using constrained wavelengths in the fitness calculation. See Table 4 for the details of the runs with constrained wavelengths

the lines marked in upper panel of Fig. 8, the implementation of this is discussed below. The “fixed” wavelength decomposition³ (see results in Table 4) tells us additional information about the spectrum; there is an average redshift of 0.070 Å of the lines in the list from their reference position. This corresponds to a velocity of around 10 km s⁻¹. The comparison of the contributions between first and second order lines in the 1404 – 1408 Å region shows that Ga-GA can successfully decompose a real, convoluted spectrum, into meaningful components.

5. Discussion

We have presented a heuristic search algorithm for the detection and analysis of spectral lines, which is free of

³ The profiles computed are allowed to deviate from the reference wavelength by, at most 0.1 Å.

Table 4. This table contains the results of Ga-GA analysing the SUMER spectrum of Fig. 8 where the wavelengths, $\langle\lambda_G\rangle$ (Å), intensities, $\langle I_G\rangle$, and widths $\langle W_G\rangle$ (Å) are the mean values of a ten run ensemble. † indicates that, in this wavelength range, a line of Ar VIII at $\lambda = 700.245$ Å (in second order) dominates the emission, as is clear from inspection of images shown by Judge et al. (1997) but this was not given in the line list. This line *was* detected in the “blind” decomposition of Sect. 4 ($\lambda_G = 1400.558$ Å, $I_G = 0.030$ and $W_G = 0.151$ Å) with correspondingly different measurements for the two lines of S III. This result illustrates that a priori information (in this case, the line list), must be correct or erroneous results will occur. Mean standard deviations in $\langle I_G\rangle$ and $\langle W_G\rangle$ are 0.002 and 0.001 respectively

Ion	Order	λ_{ref}	$\langle\lambda_G\rangle$	$\langle I_G\rangle$	$\langle W_G\rangle$
S III†	2	1400.374	1400.449	0.019	0.406
S III†	2	1400.573	1400.648	0.027	0.408
O IV	1	1401.157	1401.232	0.692	0.162
S I	1	1401.514	1401.589	0.110	0.145
S IV	1	1402.770	1402.845	2.875	0.164
S IV	1	1404.771	1404.846	0.368	0.195
O IV	1	1404.806	1404.881	0.018	0.411
S III	2	1405.566	1405.641	0.044	0.094
S III	2	1405.643	1405.718	0.108	0.230
O III	2	1405.676	1405.751	0.092	0.002
O III	2	1405.791	1405.874	0.014	0.060
S IV	1	1406.076	1406.151	0.086	0.139
O IV	1	1407.382	1407.457	0.138	0.308
O III	2	1407.701	1407.776	0.091	0.122
O III	2	1407.709	1407.784	0.270	0.207

operator bias and robust against poor or noisy data. Data are fitted simultaneously, and not sequentially, therefore limiting the propagation of systematic errors through the procedure. Coding is simple to write and easy to use, needing minimal operator input. However, the simplicity of the GA used here places limitations on the amount of information that can be extracted from spectra. Although there is *no* practical limit to the number of parameters used in the genetic decomposition, the efficiency with which the one point cross-over operator “explores” the solution space decreases as the number of parameters increases. However, such a problem can be countered simply by using a multiple point cross-over operator (see discussion in Goldberg 1989). Such adaptations are simple to make in any GA code.

In cases where data is more poorly sampled or noisier than those examined here, convergence times may become longer than the few minutes or so typical of the examples shown. It is clear from the CPU times (T_{CPU}) given in Tables 1 and 2 that although Ga-GA is not as “fast” as CURVEFIT we can see that the user must compromise between run time and the degree of accuracy required since Ga-GA has clearly demonstrated its usefulness in the presence of quite severe noise. Presumably there is also a trade-off between poorer sampling (i.e. fewer points) saving on floating point operations, and noisier data leading to many

more fitting attempts. Monitoring the convergence of the GA in the cases examined here indicates that it is adept at rapidly fitting the large scale spectral features, and progressively slower at smaller scales. This cascading nature is central to the operation of a GA, and underpins its stability in the face of noisy data (the noise being on the smallest scale is fitted last). Increasing the scale of the computation is straightforward since the generation of each child is an independent calculation (strictly, the generation of each pair of derived strings), and so the algorithm lends itself naturally to parallelisation. It is also clear that a GA routine like Ga-GA could form part of a suite of line analysis codes, with the GA offering a best initial estimate of the profile for more conventional processing methods which require a “good” initial guess.

Acknowledgements. We acknowledge the support of finance from the UK PPARC (Research grants to JCB, DAD and JI, Studentship - SMC, Visitor Grant - PGJ) and software/hardware supplied by the STARLINK project. JCB and SMC also acknowledge the support of the HAO Visitor program. SMC would like to thank PC for introducing him to the freely available PIKAIA Genetic Algorithm⁴. We would also like to thank Dr. B. Plez for helpful comments and suggestions made about this text during the refereeing process.

References

- Brekke P., Kjeldseth-Moe O., Brynildsen N., et al., 1997, Solar Phys. 170, 163
- Brynildsen N., 1994, Profile fitting to CDS/SUMER data: CDS software note No. 21, Technical report, University of Oslo
- Charbonneau P., 1995, ApJS 101, 309
- Charbonneau P., Knapp B., 1996, A User’s Guide to PIKAIA 1.0: NCAR Technical Note 418+IA
- Darwin C., 1859, On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life, J. Murray: London
- Davis L., 1991, Handbook of Genetic Algorithms. Van Nostrand Reinhold
- Diver D.A., 1995, Eur. J. Phys. 16, 211
- Diver D.A., Ireland D. G., 1997, Nucl. Instr. Meth. A 399, 414
- Goldberg D.A., 1989, Genetic Algorithms in Search of Optimization and Machine Learning. Addison Wesley
- Harrison R.A., Sawyer E.C., Carter M.K., et al., 1995, Solar Phys. 162, 233
- Holland J.H., 1962, J. Assoc. Comp. Mach. 3, 297
- Judge P.G., Hansteen V., Wikstol O., et al., 1997, ApJ (in press)
- Laming J.M., Feldman U., Schuehle U., et al., 1997, ApJ 485, 911
- Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P., 1992, Numerical Recipes in Fortran: Second Edition. Cambridge University Press
- Seely J.F., Feldman U., Schühle U., et al., 1997, ApJ 484, 87
- Wilhelm K., Curdt W., Marsch E., et al., 1995, Solar Phys. 162, 189

⁴ <http://www.hao.ucar.edu/public/research/si/pikaia/pikaia.html>