

UNCORRECTED PROOF!

A HYBRID ALGORITHM FOR SPECTRAL ANALYSIS

J. FEDERICO RAMÍREZ* and OLAC FUENTES

*Instituto Nacional de Astrofísica, Óptica y Electrónica, Luis Enrique Erro # 1, Santa María
Tonantzintla, Puebla, México*

(* author for correspondence, e-mail: framirez@ccc.inaoep.mx, Fax: 2222663152)

(Received; accepted)

Abstract. In this paper we present a method that combines evolution strategies (ES) and standard optimization algorithms to solve the problem of fitting line profiles of stellar spectra. This method provides a reliable decomposition and a reduction in computing time over conventional algorithms. Using a stellar spectrum as input, we implemented an evolution strategy to find an approximation of the continuum spectrum and spectral lines. After a few generations, the parameters found by ES are given as starting search point to a standard optimization algorithm, which then finds the correct spectral decomposition. We used Gaussian functions to fit spectral lines and the Planck function to represent the continuum spectrum. Our experimental results present the application of this method to real spectra, showing that they can be approximated very accurately.

Keywords: evolution strategies, optimization, spectral analysis

1. Introduction

An important task in the analysis of stellar spectra is to identify and measure the flux of emission and absorption lines. Standard spectral decomposition techniques using non-linear least squares fitting algorithms such as the Levenberg-Marquardt method (Levenberg, 1944; Marquardt, 1963) or unconstrained optimization methods such as simplex search of Nelder and Mead (1965) have been used to fit line profiles with spectral profiles of ionic and atomic transitions of known wavelengths; however, these techniques proved to be dependent on initial parameters provided by the user (McIntosh et al., 1998) and often required very high computing times. We propose a hybrid algorithm to deal with these difficulties. The method first uses an evolution strategy (ES) to find an approximate solution to the problem, then, this solution is used as the starting search point by a standard optimization algorithm. Since the initial parameters are near the optimum, the algorithm converges quickly, and since human intervention is reduced, it has the potential of limiting or eliminating user biases.

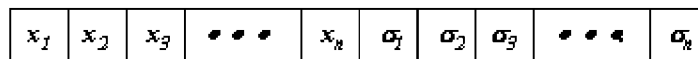


Figure 1. Representation of an individual.



Experimental Astronomy **0**: 1–18, 2003.

© 2003 Kluwer Academic Publishers. Printed in the Netherlands.

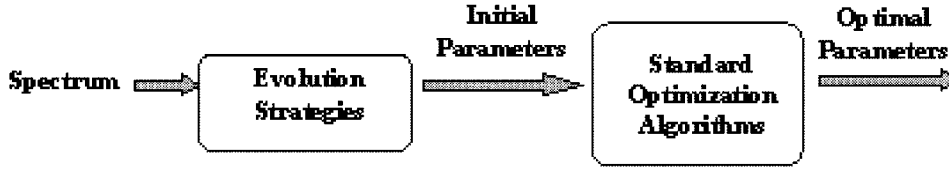


Figure 2. The hybrid system.

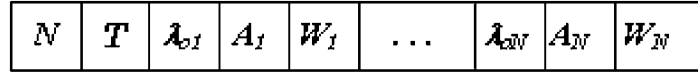


Figure 3. Representation of the object variables in an individual.

In this work, spectral lines are fitted by a summation of Gaussian functions, while the continuum is approximated by the Planck function. The free parameters of these functions are encoded as individuals in the ES, which initially assigns random values to them and after running for few generations finds a set of values for the parameters near the solution. Then, standard optimization algorithms (Simplex search and Levenberg-Marquardt algorithms) use these parameters as initial search point to find the best set of values for fitting the model to the spectrum.

The remainder of this paper is organized as follows: Section 2 gives a brief overview of evolution strategies. Section 3 presents a brief description of stellar spectra. Section 4 presents the method used in our experiments, Section 5 presents experimental results and discussion and Section 6 presents conclusions and outlines directions for future work.

2. Evolution strategies

Evolution Strategies (ES) (Schwefel, 1975; Bäck, 1996) are a class of probabilistic search algorithms loosely based on biological evolution that have been applied successfully to optimization problems in poorly-modelled domains and in the presence of noisy data (Fuentes and Nelson, 1998). They work on a population of individuals, where each of them represents a search point in the space of potential solutions to a given problem. In evolution strategies, a vector of real numbers represents an individual. This is a good representation when the problem at hand deals with continuous parameters. Each individual is formed by a vector of elements called *object variables* x_i , and each variable has associated to it a standard deviation called *strategy parameter* σ_i , as shown in Figure 1.

Initially, the algorithm randomly generates a population of individuals; subsequently this population is updated by means of randomized processes of recombination, mutation, and selection. Each individual is evaluated according to a fitness function that depends on the problem to be solved.

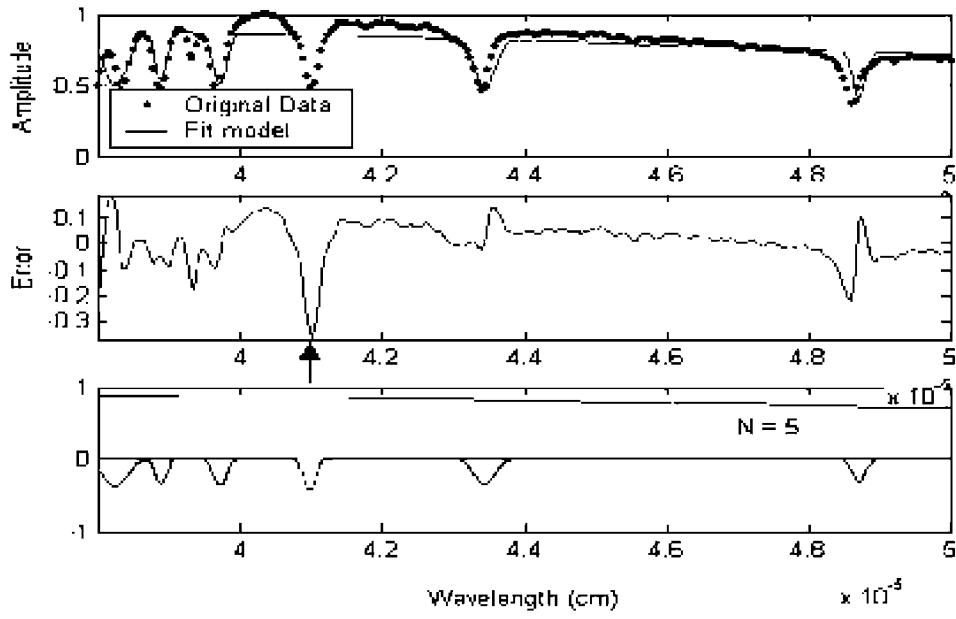


Figure 4. Intelligent-mutation.

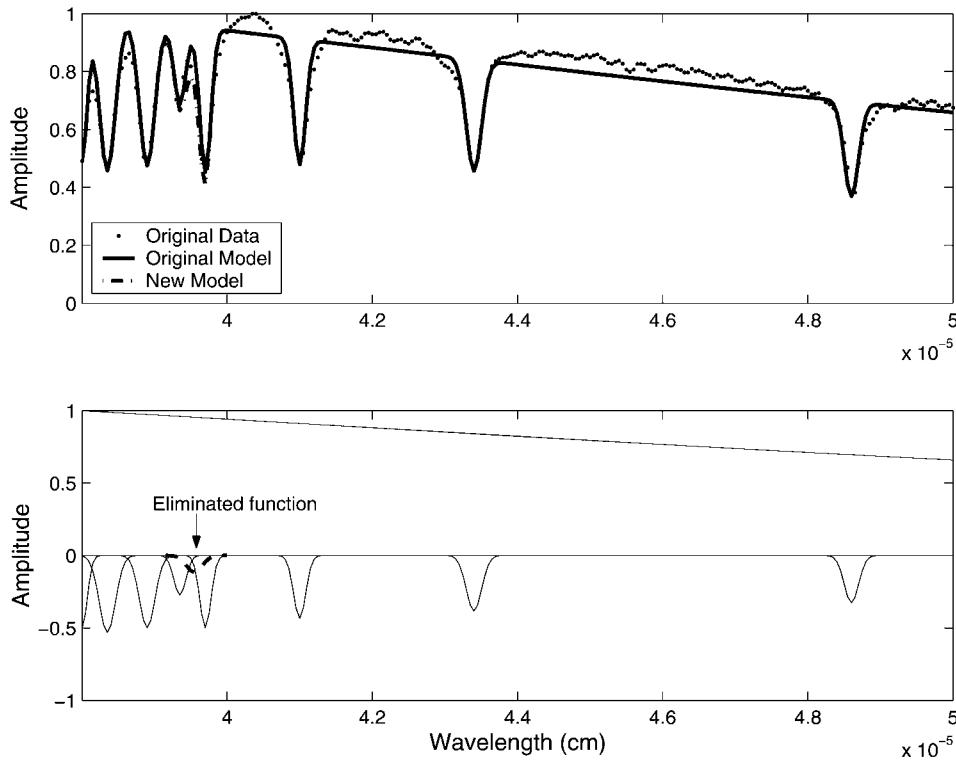
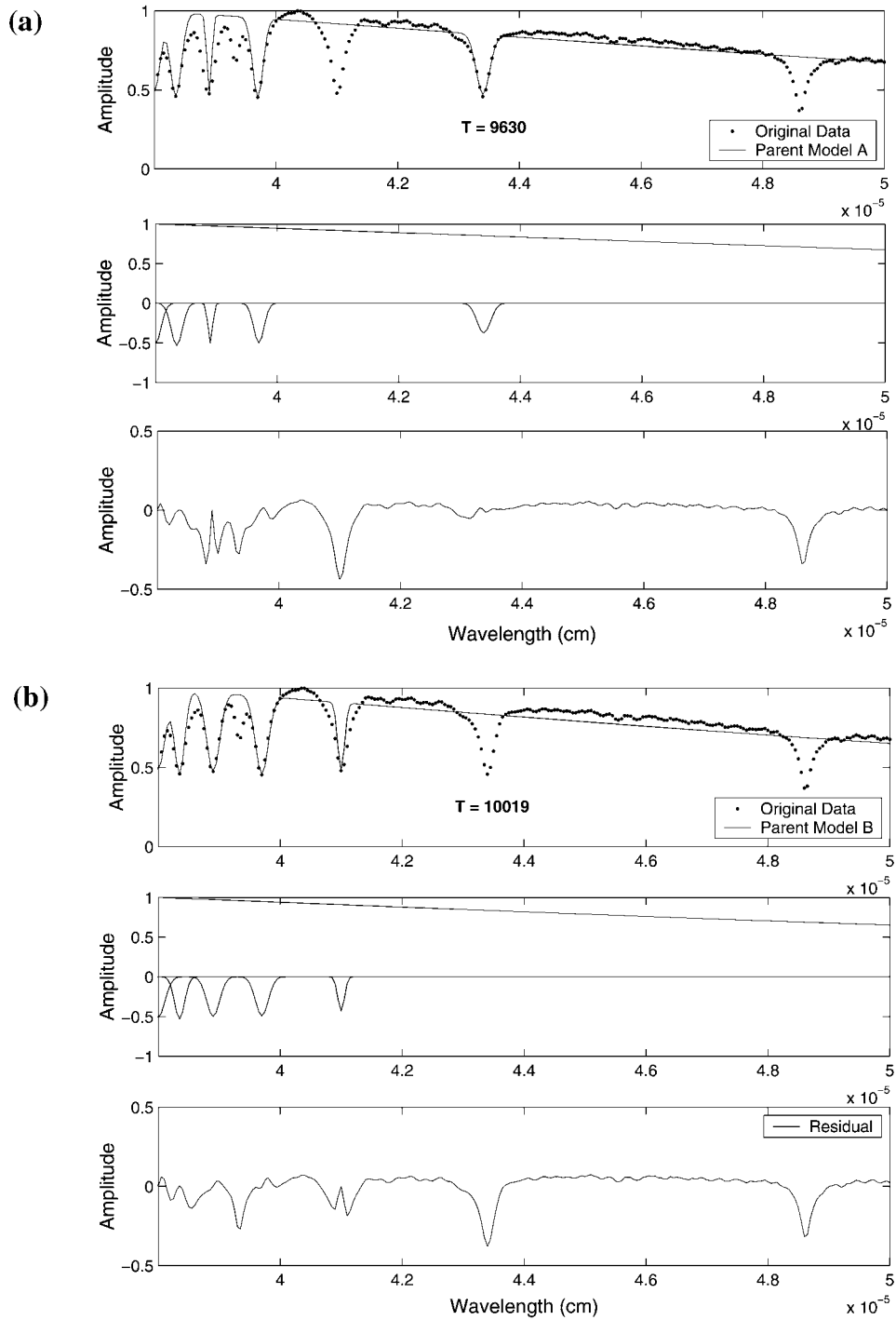


Figure 5. Elimination-mutation.



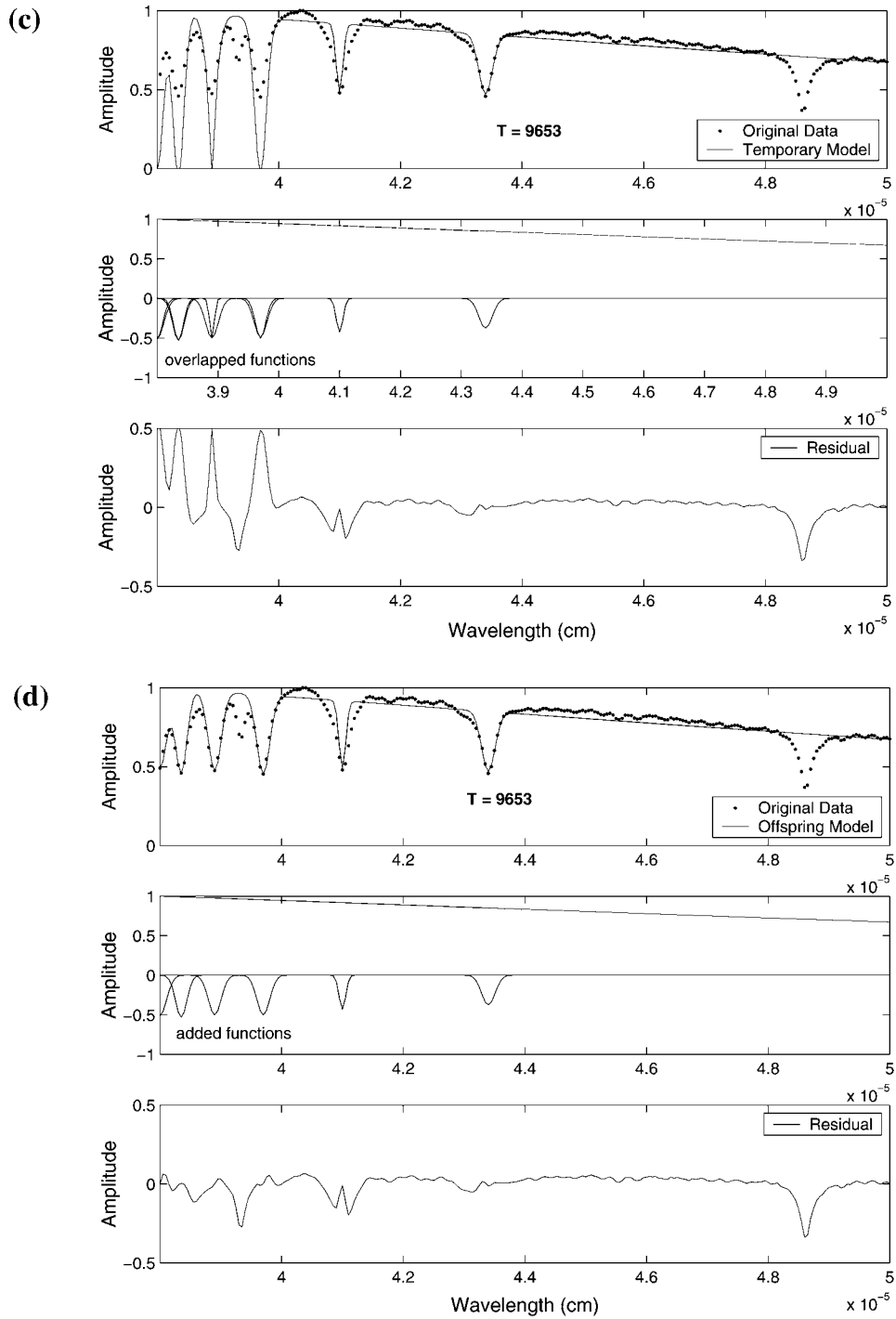


Figure 6c-d. Recombination operator. (c) Temporary Model, (d) Offspring Model.

The selection process favors the fit individuals from the current population to reproduce in the next generation. In evolution strategies, this process is completely deterministic. In $(\mu + \lambda)$ -selection, the μ best individuals from the union of μ parents and λ offspring are selected to form the next parent generation, while in (μ, λ) -selection this operator selects the μ best individuals from the λ offspring only; for this $\mu < \lambda$ is required.

The recombination process allows to combine information from different members of a population, creating offspring from them. Bäck (1996) shows a variety of recombination mechanisms used in evolution strategies. Typical examples of them are: discrete recombination, which is a sexual operation that creates two offspring vectors from two parent vectors copying selected elements from each parent; and intermediate recombination, which is commonly used as an arithmetic average with some variants. These operators can be used in sexual or panmictic form. In the sexual form, every element of an offspring is the result of recombination between two individuals randomly chosen from the parent population. In panmictic form, each element of an offspring may be the result of recombination among one individual and several other individuals randomly chosen from the parent population.

Mutation is an asexual operator that generates random changes to an individual and often provides new relevant information. The mutation operator is applied independently to each object variable of an individual. It is carried out as shown in Equation (1). The strategy parameters may be mutated using a multiplicative, logarithmic normally distributed process as shown in Equation (2).

$$x'_i = x_i + \sigma_i \cdot N(0, 1) \quad (1)$$

$$\sigma'_i = \sigma_i \cdot \exp(N(0, 1) + N_i(0, 1)) . \quad (2)$$

where $N(0, 1)$ is a normally distributed random variable having an expectation of zero and a standard deviation of one, $N_i(0, 1)$ indicates that the random variable is sampled anew every time the index i changes.

Rechenberg proposed a deterministic adjustment of strategy parameters during evolution, called the 1/5-success rule (Rechenberg, 1973), which reflects that, on average, one out of five mutations should cause an improvement in the objective function values to achieve best convergence rates. If more than 1/5 of the mutations are successful, σ is increased, otherwise it is decreased.

3. The methods

A drawback of standard optimization algorithms is that they are dependent on the initial values to find the optimal parameters, so these values must be close to the solution. The hybrid algorithm we propose shown in Figure 2, can overcome this problem, using ES to find the input values to a standard optimization algorithm.

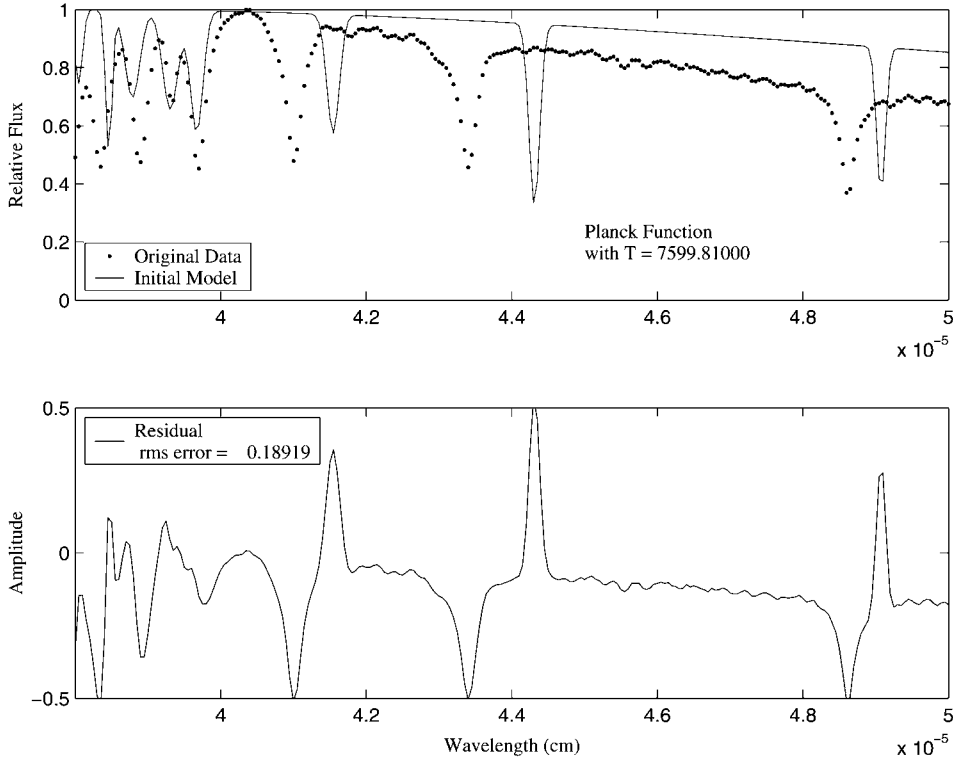


Figure 7. Initial fitting model given to the Simplex and Levenberg-Marquardt algorithms on the digital spectrum of a star of type A8.

The model used to fit the stellar spectra is defined by $P(\lambda)$ in Equation (3), where $F(\lambda)$ is the background and the summation of $G_i(\lambda)$ is a set of Gaussian functions representing the stellar lines superimposed on the background. The first function $F(\lambda)$ used in our model is the Planck function, presented by Equation (4), which approximates a blackbody emission and has the Temperature of the star as free parameter.

$$P(\lambda) = F(\lambda) + \sum_{i=1}^n G_i(\lambda) \quad (3)$$

$$F(\lambda) = \frac{2hc}{\lambda^5} \cdot \frac{1}{\exp\left(\frac{hc}{\lambda kT}\right) - 1}, \quad (4)$$

where h is the Planck constant, k is the Boltzman constant, c is the speed of light and T is the temperature of the star.

Each Gaussian function is described by three parameters: a center point (λ_0), a variance (σ) and an amplitude (A), as defined by Equation (5). These functions

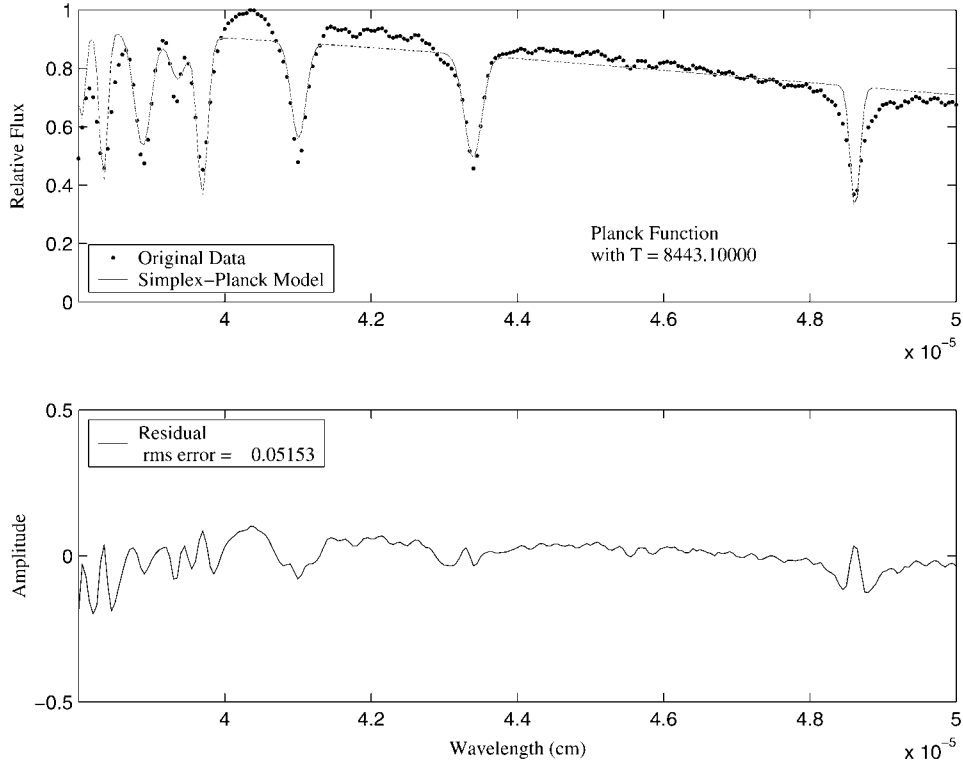


Figure 8. Final fitting model generated by the Simplex algorithm on the digital spectrum of the star of type A8 after 20 000 model evaluations.

must be spread over the range of the evaluation data and each one must be assigned with an appropriate variance and central point in order to cause the best fitting to the spectral lines.

$$G_i(\lambda) = A \cdot \exp\left(-\frac{(\lambda - \lambda_0)^2}{2\sigma^2}\right). \quad (5)$$

The free parameters of $P(\lambda)$ are encoded as object variables of the individuals used by the ES. The representation of the object variables of an individual for this problem has the form shown in Figure 3, where n_g corresponds to the number of Gaussian functions, and $T, \lambda_{0i}, A_i, \sigma_i$ are the free parameters of $P(\lambda)$.

In our evolution strategy implementation, called *GaES-Planck*, we added an operator called intelligent-mutation. This procedure checks for the wavelength with the greatest difference between data generated by the fitting model and the original data, and adds a Gaussian function in this position with the same amplitude as the error size and a random variance value, as shown in Figure 4. Similarly, we added another operator called elimination-mutation, which carries out the opposite func-

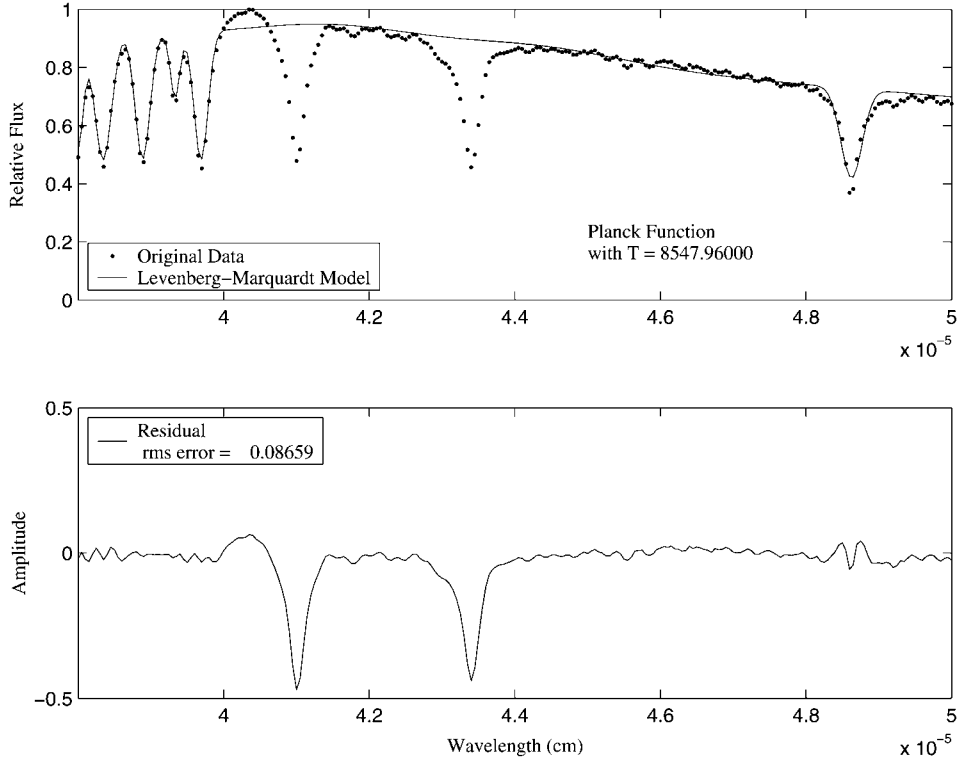


Figure 9. Final fitting model generated by the Levenberg-Marquardt algorithm on the digital spectrum of the star of type A8 after 3509 model evaluations

tion; it suppresses the Gaussian function that produces the smallest error reduction, as shown in Figure 5.

The recombination operator implemented in this algorithm creates an offspring model from two parent models randomly chosen from the parent population. It applies the standard intermediate recombination on the temperature parameters of the parent models using Equation (6), where β is a random number obtained from a uniform distribution in the $[0, 1]$ range. In order to update the Gaussian functions, this procedure merges the parent models into a temporary model and removes overlapping functions. Then, it adds a Gaussian function in the wavelength with the greatest error between the original data and the fitting model within the wavelength interval covered by the removed functions. The amplitude and the base of the added function correspond to the error size and the wavelength interval covered by the removed functions as shown in Figure 6.

$$T_{\text{offspring}} = \beta * T_{\text{parent A}} + (1 - \beta) * T_{\text{parent B}} . \quad (6)$$

The fitness function of ES is shown in Equation (7), where n_d is the number of points in the digital spectrum, $P(\lambda_i)$ is a point generated by the model at λ_i , $f(\lambda_i)$

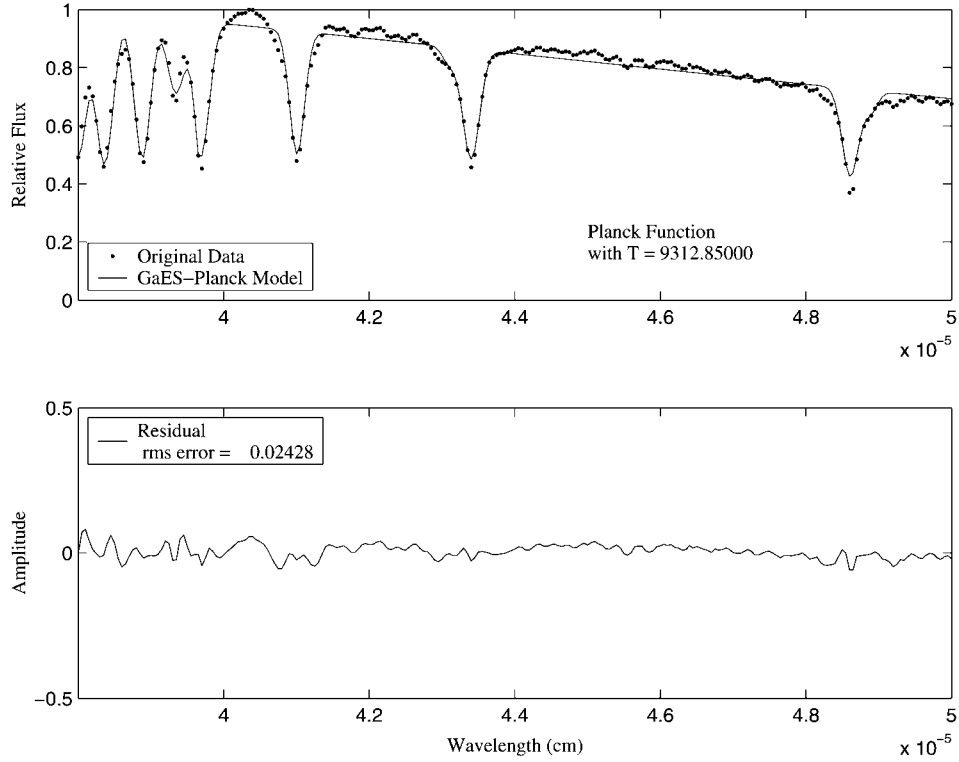


Figure 10. Fitting model generated by the GaES-Planck algorithm after 20 000 model evaluations.

is a point in the original data, n_g is the number of Gaussian functions, and α is a constant. The first term of the fitness function corresponds to the root mean squared error and the second term is a penalty, which favors individuals with fewer Gaussian functions.

$$\text{Fitness} = \sqrt{\frac{\sum_{i=1}^{n_d} (P(\lambda_i) - f(\lambda_i))^2}{n_d}} + \alpha \cdot n_g . \quad (7)$$

In our experimental results we show that the GaES-Planck can find a good model by itself, but this is achieved after a large number of generations. Also, we show that the combination of GaES-Planck and standard optimization techniques decreases the computation time and achieves very good accuracy.

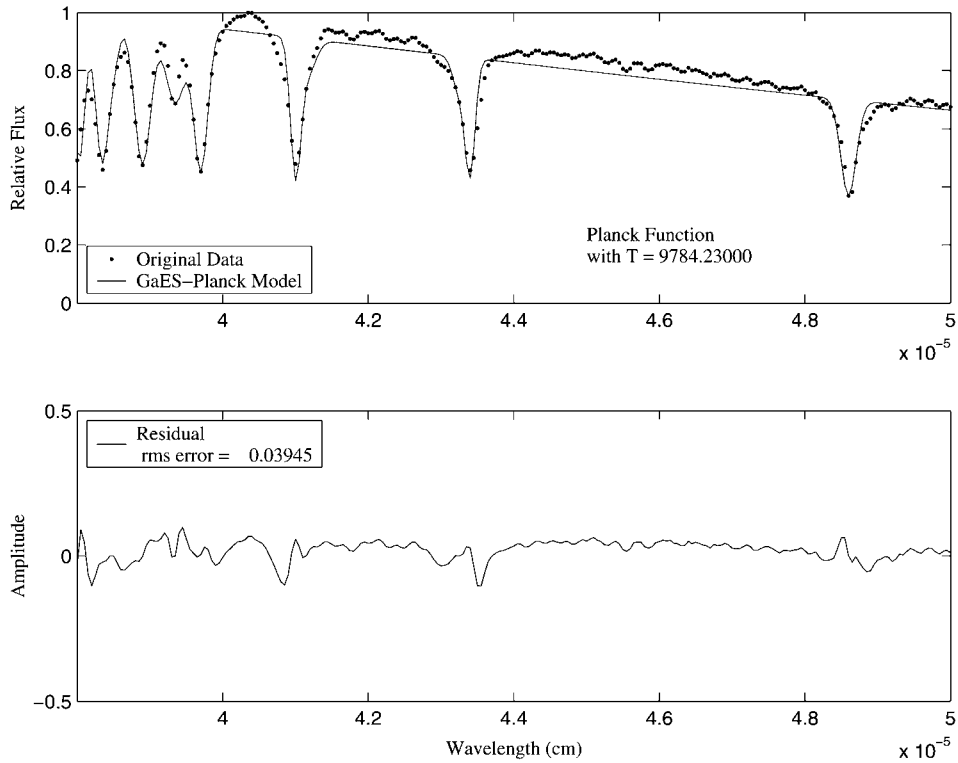


Figure 11. Fitting model generated by the GaES-Planck algorithm after 3500 model evaluations.

4. Experimental results

In this section we detail the results of combining the GaES-Planck algorithm with the Levenberg-Marquardt algorithm and with the Simplex search algorithm using the rms error of the fitting and the number of model evaluations as comparison measures. This method was applied to a real digital optical spectrum obtained from Davis' database (Silva, 1992). We used for our experiments the real digital optical spectrum of a star of type A8V in the 380–500 nm wavelength range.

For comparison, first we ran the Simplex direct search and Levenberg-Marquardt algorithms from the Matlab Optimization Toolbox setting the maximum number model evaluations to 20 000. Then, we gave an initial model with 8 line profiles close to the spectral lines of the reference spectrum, as shown in Figure 7. The Simplex algorithm generated the model shown in Figure 8 after 20 000 model evaluations, while the Levenberg Marquardt algorithm generated the model shown in Figure 9 after 3509 evaluations. The latter algorithm stops earlier because the gradient in the search direction becomes too small. The rms error achieved by the first method is 0.0515 while the rms error achieved by the second method is 0.0866.

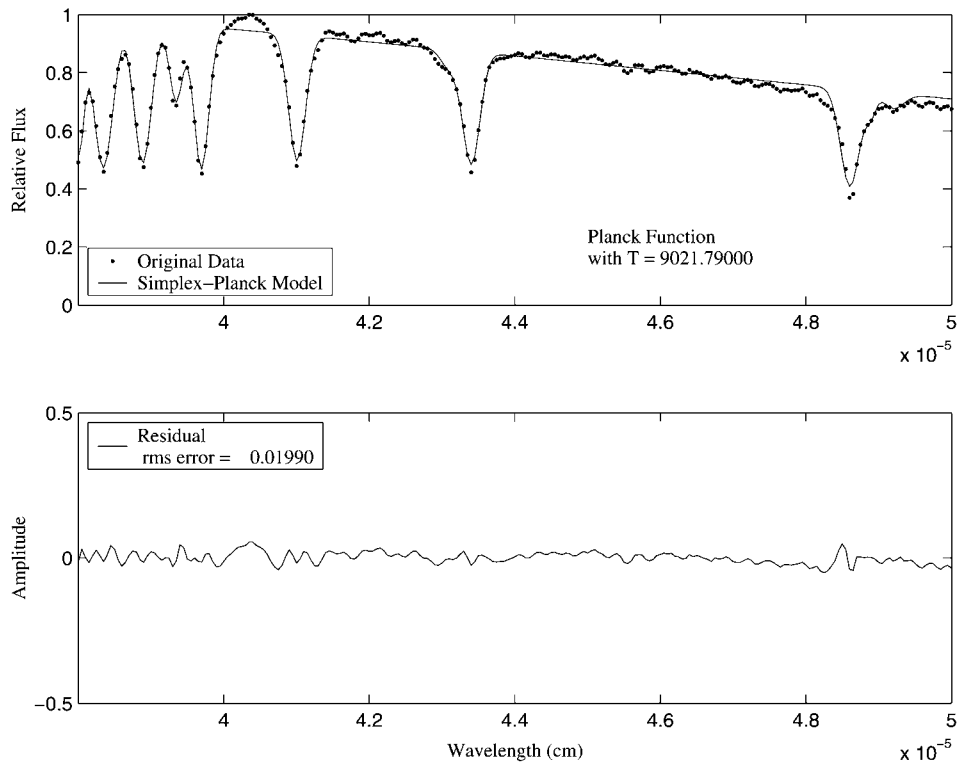


Figure 12. Fitting model generated by the GaES-Planck algorithm after 100 000 function evaluations.

In a second experiment, we ran the GaES-Planck algorithm for several generations until it performed 3500 model evaluations as the Levenberg-Marquardt method did in the previous experiment. The algorithm generated the model shown in Figure 11 with an rms error of 0.03945. The object variables of the initial population of GaES-Planck consist of only two fields, N and T. GaES-Planck assigns a zero value to N, and a random number in the $[0, 1]$ range to T. Then, we ran the algorithm until it performed 20 000 evaluations as the Simplex method did in the previous experiment. Figure 10 shows the model generated by the algorithm achieving an rms of 0.02428. Also, we ran the algorithm until it reached an rms error of less than 0.0200 in order to measure the number of model evaluations. The algorithm carried out 100 000 model evaluations achieving an rms error of 0.0199 as shown in Figure 12. In this experiment a population of 50 individuals was established for the GaES-Planck algorithm. In addition, we applied the standard mutation operator on 70% of the parent population, and each of the other GaES-Planck operators on 10%.

In a third experiment we combined the Simplex and Levenberg-Marquardt algorithms with the GaES-Planck, constraining the latter algorithm to run until it had not improved upon its fitness function for 20 generations, or until 100 generations

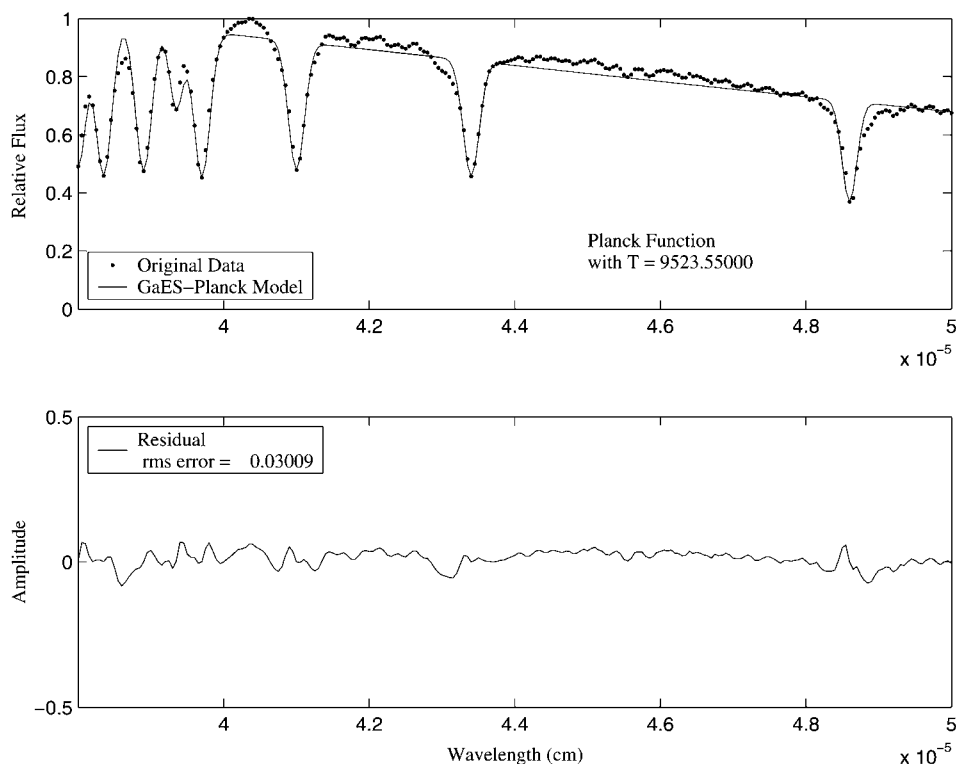


Figure 13. Fitting model given to the standard optimization algorithms by the GaES-Planck algorithm after 5000 model evaluations with a population of 50 individuals.

were reached. Also, we set the maximum number of model evaluations allowed in the standard optimization methods to 2000. First, we ran the hybrid algorithm with a population of 50 individuals per generation, that is, the algorithm was constrained to perform 7000 model evaluations. Figure 13 shows the model with 8 Gaussian functions passed to the Simplex and Levenberg-Marquardt algorithms after 5000 model evaluations by the GaES-Planck algorithm, while Figures 14 and 15 show the final results achieved by the standard optimization algorithms. The GaES-Planck-Simplex method achieved an rms error of 0.01985 and performed a total of 7000 model evaluations, while the GaES-Planck-Levenberg-Marquardt method achieved an rms error of 0.01973 and performed a total of 5521 model evaluations.

Due to the fact that the GaES-Planck algorithm does not need to give an accurate model to the standard optimization algorithm, we reduced the population of the hybrid algorithm to 10 individuals per generation in order to decrease the computing time. We used the same values for the GaES-Planck operator rates as in the second experiment, (the algorithm applied the standard mutation to 7 individuals of the population per generation and the remaining operators on one individual each). The algorithm obtained similar fitting models to those of the previous experiment,

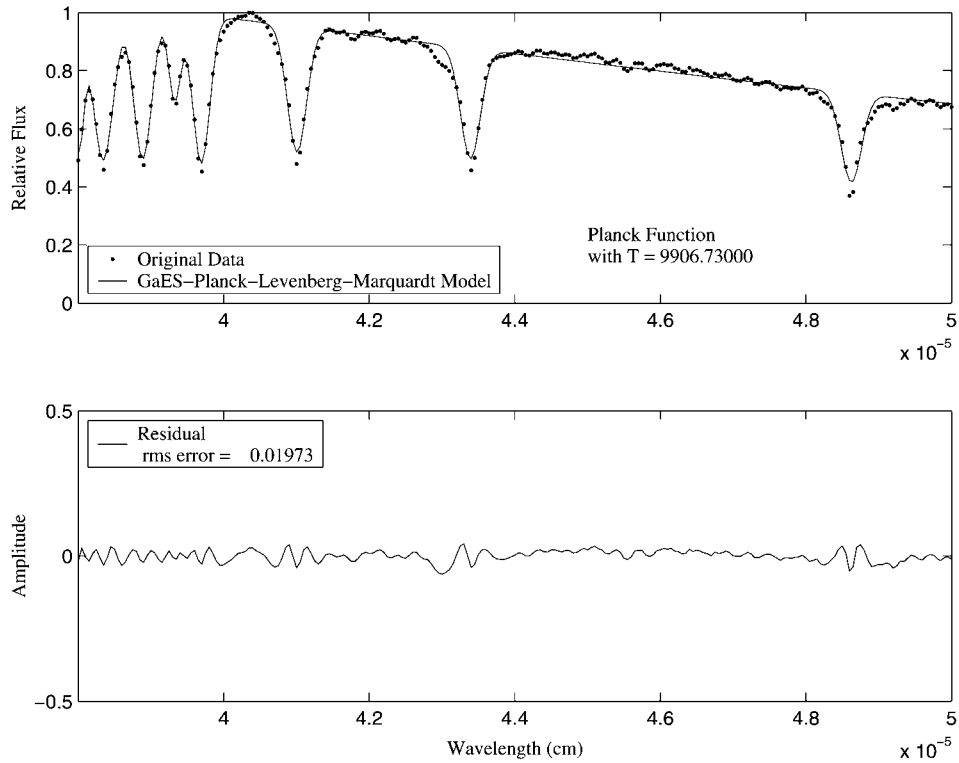


Figure 14. Final fitting model generated by the GaES-Planck-Levenberg-Marquardt algorithm on the digital spectrum of the star of type A8 after 5521 model evaluations and a population of 50 individuals.

however, the number of model evaluations was greatly reduced. The GaES-Planck-Simplex method achieved an rms error of 0.0200 and performed 3000 model evaluations while the GaES-Planck-Levenberg-Marquardt method achieved an rms error of 0.01973, after 1522 model evaluations. The models generated in this experiment are shown in Figures 16 and 17.

In order to analyze how sensitive the results of the final fit are to the variation of the hybrid algorithm operators rates, we ran the GaES-Planck-Simplex and the GaES-Planck-Levenberg-Marquardt algorithms with different values of the evolution operation rates. A population of 50 individuals was established in this experiment varying the intelligent-mutation and elimination mutation rate from 0.02 to 0.20 while keeping the recombination rate in 0.10, (i.e. the algorithm applied the intelligent-mutation as well as elimination-mutation on 1 to 10 individuals of the population, and their recombination operator to 5 individuals). Also, we varied the recombination rate from 0.10 to 0.40, keeping the intelligent-mutation and the elimination-mutation rates at 0.04, (i.e. 2 individuals per generation each). The final fitting results did not change significantly from the previous experi-

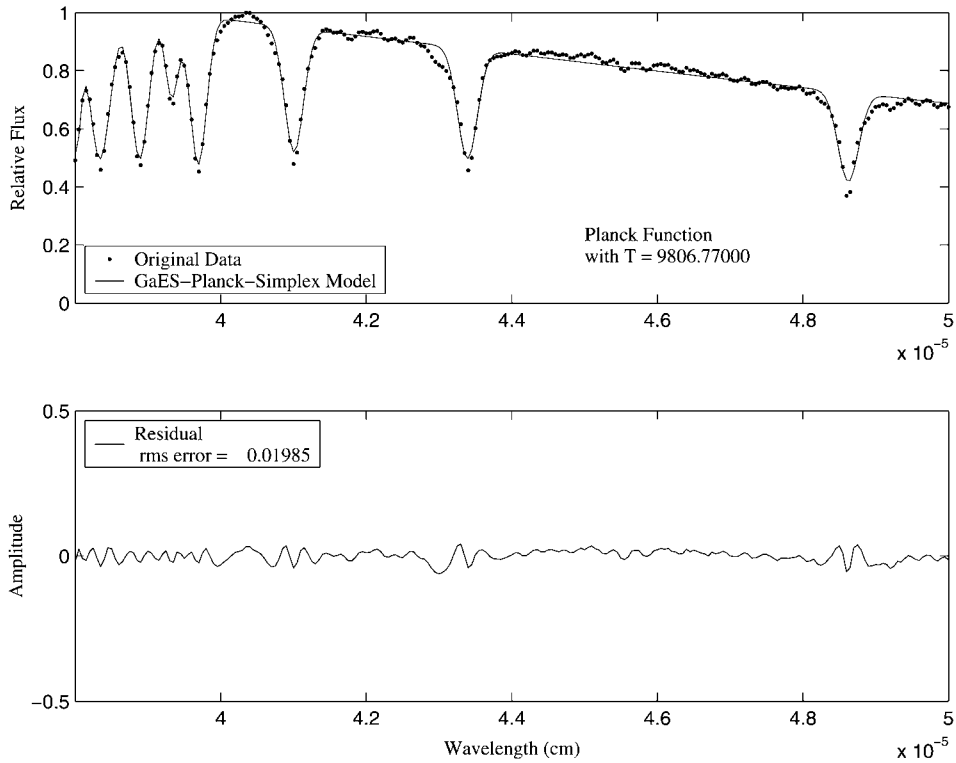


Figure 15. Final fitting model generated by the GaES-Planck-Simplex algorithm on the digital spectrum of the star of type A8 after 7000 model evaluations and a population of 50 individuals.

ment using the hybrid algorithm. The GaES-Planck-Simplex algorithm carried out 7000 model evaluations achieving an rms error of 0.0200 while the GaES-Planck-Levenberg-Marquardt algorithm performed 5430 model evaluations with an rms error of 0.01973. This shows that the algorithm is not very sensitive to these parameters.

In Table I we show a comparison of the different methods used in our experiments. In this table we observe that using only GaES-Planck gives a good result, but it requires a large computing time. If we combine the GaES-Planck algorithm with a standard optimization algorithm in a hybrid method, it will give a better result and will reduce the computational time. Taking the Simplex and Levenberg-Marquardt algorithms as reference, the GaES-Planck-Simplex algorithm reduced the rms error by 61% and the computing time by 85%, while the GaES-Planck-Levenberg-Marquardt algorithm reduced the rms error by 77% and the computing time by 56%.

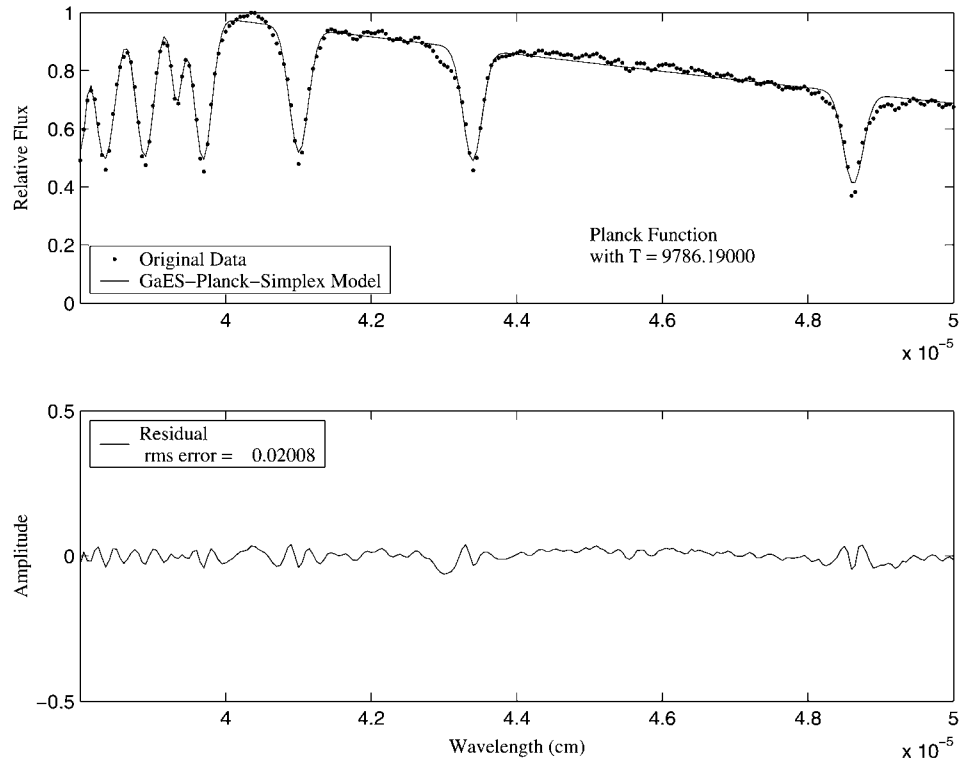


Figure 16. Final fitting model generated by the GaES-Planck-Simplex algorithm on the digital spectrum of the star of type A8 after 3000 model evaluations and a population of 10 individuals.

TABLE I

Comparison of the different methods for spectral analysis with populations of 50 and 10 individuals

	Population size	Fitting RMS error	Number of model evaluations
Simplex	–	0.0513	20000
Levenberg-Marquardt	–	0.0866	3504
GaES-Planck	50	0.0243	20000
GaES-Planck	50	0.0395	3500
GaES-Planck	50	0.0199	100000
GaES-Planck-Simplex	50	0.0198	7000
GaES-Planck-Levenberg-Marquardt	50	0.0197	5521
GaES-Planck-Simplex	10	0.0200	3000
GaES-Planck-Levenberg-Marquardt	10	0.0197	1522

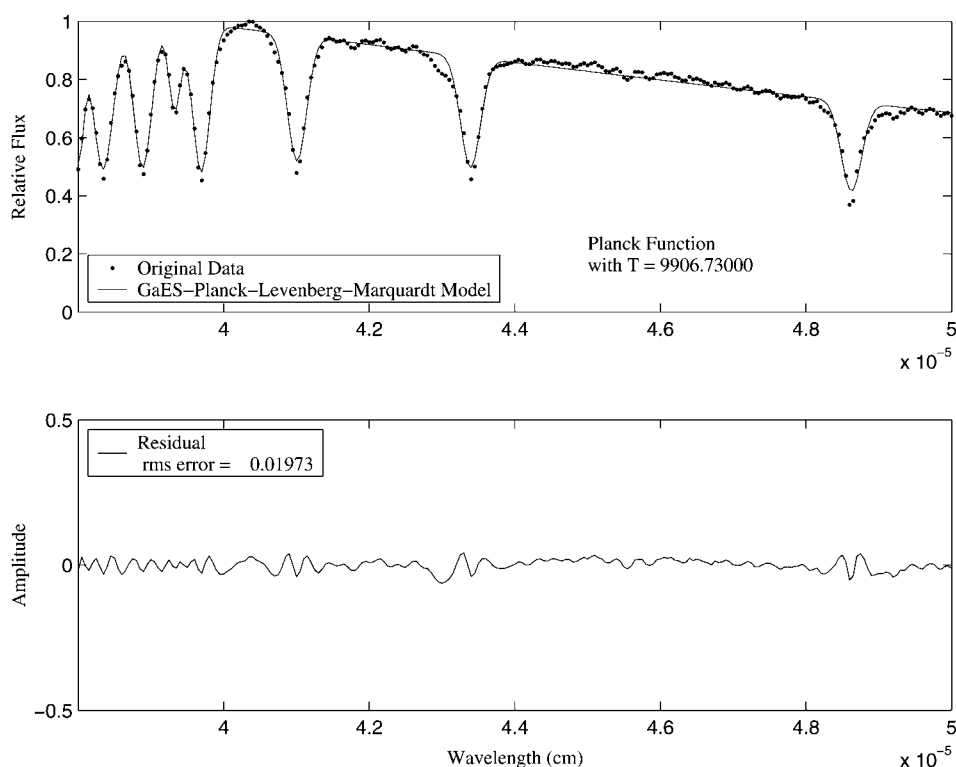


Figure 17. Final fitting model generated by the GaES-Planck-Levenberg-Marquardt algorithm on the digital spectrum of the star of type A8 after 1522 model evaluations and a population of 10 individuals.

5. Conclusions

In this paper we have presented a hybrid algorithm to fit a model of Gaussian functions and the Planck function to digital spectra in order to find spectral lines and the continuum using ES and standard optimization algorithms. Our experimental results show that applying this method to a digital stellar spectrum provides the following advantages over conventional algorithms:

- It does not require user input regarding initial search parameters nor the number of Gaussian functions;
- It decreases the computing time.
- It yields more accurate results than the Simplex and Levenberg-Marquardt algorithms alone.

Future work will extend the experimental results to more spectra and compare the results with theoretical models.

References

- Bäck, T.: 1996, *Evolutionary Algorithms in Theory and Practice*, Oxford University Press.
- Fuentes, O. and Nelson, R. C.: 1998, 'Learning Dexterous Manipulation Strategies for Multifingered Robot Hands Using the Evolution Strategy', *Machine Learning* **31**, 223–237.
- Levenberg, K.: 1944, 'A Method for the Solution of Certain Problems in Last Squares', *Quart. Appl. Math.* **2**, 164–168.
- Marquardt, D.: 1963, 'An Algorithm for Least Squares Estimation of Nonlinear Parameters', *SIAM J. Appl. Math.* **11**, 431–441.
- McIntosh, S. W., Diver, D. A., Judge, P. G., Charbonneau, P., Ireland, J. and Brown, J. C.: 1998, 'Spectral Decomposition by Genetic Forward Modelling', *Astronomy and Astrophysics Supplement Series* **132**, 145–153.
- Nelder, J. A. and Mead, R.: 1965, 'An Algorithm for Least Squares Estimation of Nonlinear Parameters', *Computer J.* **7**, 308–313.
- Rechenberg, I.: 1973, *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*, Frommann-Holzboog Verlag, Stuttgart.
- Schwefel, H.-P.: 1975, 'Evolutionsstrategie und Numerische Optimierung', *Dissertation*, Technische Universität Berlin.
- Silva, D. R.: 1992, 'A New Library of Stellar Optical Spectra', *Astrophysical Journal Supplement Series* **81**, 865–881.

Author, please supply some new (electronic) figures!