An Application of Heuristic Algorithms to Radial Velocity Data from Multiple-Planet Extrasolar Systems

M. Baloković [1], *M. Kürster* [2]

[1]Department of Physics, Faculty of Sciences, University of Zagreb, Bijenička 32, 10000 Zagreb, Croatia
[2]Max Planck Institute for Astronomy, Königstuhl 17, D-69117 Heidelberg, Germany
mislavb@fizika.org

We present our work on the application of heuristic global optimization algorithms to radial velocity data from multi-planet extrasolar systems. The aim is to find an optimum fit for the whole system simultaneously, thus avoiding any assumptions about the correctness of previously found planets. The complexity of the task calls for algorithms efficient in global optimization problems with many degrees of freedom. We explored the capabilities of the basic genetic and simulated annealing (Metropolis) algorithms and added more features to improve their performance. They were then combined for optimum performance on this specific problem, utilizing the morphology of the parameter space in a new search strategy. Here we present the programs and their application to artificial and real radial velocity data in which multiple planets were previously identified.

Introduction

Measurements of stellar radial velocities are by far the most productive method for finding extrasolar planets. A standard time series analysis method is the Lomb-Scargle periodogram [7, 8]. It was shown by Scargle that the Lomb-Scargle periodogram is equivalent to least-squares or chi-square fitting of sinusoidal functions. Multiple-planet systems are usually identified by successive reduction of data with radial velocity curves of proposed planets whose periods are found by the periodogram.

An obvious drawback of the sequential approach is that it needs to fix the period of one planet before even searching for the next one. An optimum radial velocity curve for the full multiple planet system might be somewhat different from what is found by successive single planet fits. Recently several researchers (see e.g. [4]) have made use of global optimization techniques to check this possibility, but no one has so far reported on the efficiency of such programs.

A simultaneous fit of multiple Keplerian radial velocity curves is a global optimization problem in a complex parameter space dominated by deep and narrow minima. As the function to be minimized we choose the chi-square $\left(\chi^2\right)$ statistic as a measure of the goodness of fit. In our case it is a function of $5N+1$ parameters, where N is the number of planets in the model. The parameter search space can be reduced to $4N$ by solving a set of $N+1$ linear equations at each function evaluation. Also, for every complete set of parameters, $2M$ transcendental equations need to be solved, where M is the number of data points. Hence, evaluation of each possible solution has a considerable computer time cost. Our goal was to investigate if heuristic optimization would solve this problem efficiently.

Many standard optimization procedures, mostly local, or limited to cost functions which are well-behaved, are not well suited to solve this problem in reasonable time. The algorithms we used, the simulated annealing algorithm (SA, also called the Metropolis algorithm) and the genetic algorithm (GA), use principles we find in nature and know that they are working. Since the only proof of their success is human experience, they are called heuristic, or intuition-based. Indeed, despite the fact that there is no rigorous mathematical proof supporting their use, they were shown to be extremely useful in many applications in astrophysics as well as in other sciences. For a general introduction to these algorithms, refer to [5] and [2]. Some specifics of our adaptation will be presented below.

Strategy

Our programs were first written in Fortran 90 and later combined with Mathematica 6.0 by Wolfram Research. Several aspects had to be solved prior to choosing an algorithm to search the parameter space for a feasible orbital solution. We investigated options of algebraically solving for some orbital parameters, concluding that this could be done only for radial velocity amplitudes and the zero-point offset. Period, time of periastron, longitude of periastron and eccentricity of each planet need to be found by other means. However, not all parameters of a Keplerian orbit are equally difficult to find. Finding periods (frequencies) is the hardest task, while finding longitude of periastron is relatively easy. This led us to a new hierarchical strategy of searching the parameter space.

There are two search algorithms involved at different levels. The outer one searches through the parameter subspace of frequencies, which has N dimensions — equal to the number of planets in the model. For each proposed N-tuple of frequencies, the inner search algorithm tries to find N pairs of eccentricities and times of periastron. These $3N$ parameters are enough for Kepler's equation to be solved (for each data point). Once solved, its results are used by a local optimization algorithm for finding the best possible longitude of periastron values. For each proposed set of N longitudes, a set of $N + 1$ linear equations need to be solved to complete the list of orbital parameters for the solution. Fitness of the solution is then evaluated by its χ^2 value, and the program returns to the second level. If no satisfactory set of eccentricities and times of periastron can be found for the chosen set of frequencies in given number of trials, then the program returns to the first level and picks another set of frequencies. The choice of search algorithms is arbitrary here, but we used SA and GA in different arrangements to examine the performance of these programs. The tests we conducted imply that the best combination is the GA+GA one. The Performance Comparison section demonstrates its efficiency compared with the other programs.

Search Algorithms

We investigated the simulated annealing (SA) algorithm, whose name comes from imitating the slow cooling of a melted metal in order to allow its atoms to arrange themselves into a crystalline rather than an amorphous arrangement. The optimum *state* for the system is the lowest-energy one, in correspondence to the lowest chi-square orbital solution in our problem. Although the solution is expected to occur at the end of a program run, it commonly appears that the state of lowest energy is encountered sometime during the program, but gets superseded by a worse one due to probability. To avoid the risk of losing such states, our program saves low chi-square states in a list and locally optimizes them at the end of the program. These states usually get improved and sometimes the global optimum is found among them. This strategy also reduces time cost, since the SA is less efficient in local optimization than some standard local optimizers. This sort of a list is usually referred to as the *tabu* list because the program is not allowed to converge to the same states twice, until it reaches the phase of tabu list examination at the end of the program. Also, it was useful to restart the program when it stayed focused on a small portion of the parameter space for too long without good results. The improved version of the program (the one we refer to as the *modified SA* in this paper) included the tabu list with local optimization at the end, and the restarting option. However, the performance of this program was inferior to the others, as demonstrated in the next section.

Genetic algorithms (GA) use the principle of evolution to advance a *population* of orbital solutions towards better parameters. Having generated a new set of solutions by mixing the old ones, a population is subjected to a selection procedure. The purpose of selection is to allow only the best solutions — those with the lowest χ^2 value — to advance to the next iteration of the algorithm. Under a constant pressure of selection, but saving some variety at the high-χ^2 end of the population, the algorithm drifts towards a more favorable pool of orbital parameters. Many options may be added to this algorithm, but the ones we have found to be the most important are the mutation rate and the selection pressure. Their influence is discussed in the next section.

Performance Comparison

The comparisons were made on artificial datasets with known orbital parameters. A test of several different program realizations is demonstrated in Figure 1. It displays the number of successful program runs completed in fixed time in arbitrary units. The last set of columns refer to runs failing to reach the optimum chi-square in given time. 30 runs on the same artificial two-planet problem for each of the following

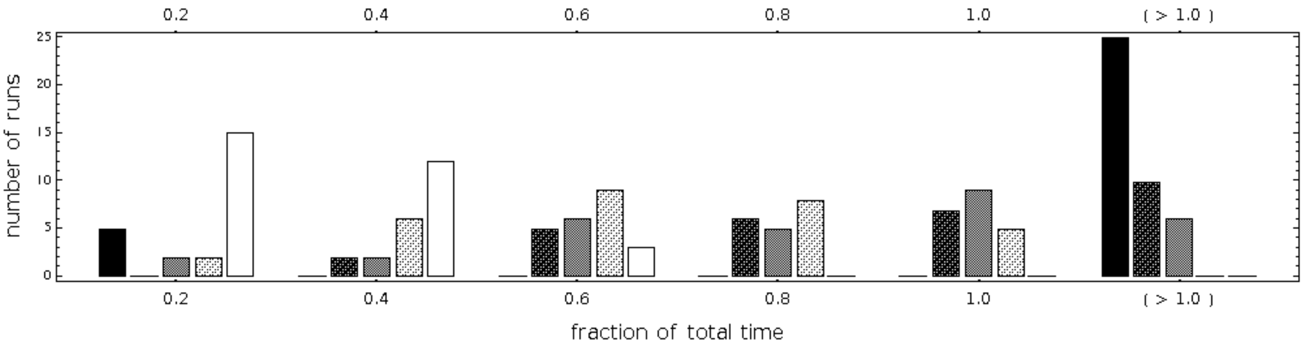

Figure 1: A performance comparison of the discussed programs. The column grayscale from dark to white represents programs in the order listed in the text below. Column height in the first five bins indicates the number of runs completed by the specified total time fraction, while the last one shows the number of runs that failed to converge in the given time interval.

programs are displayed here in the order in which they are presented in Figure 1, from the least beneficial algorithm (black column) to the most useful one (white column):

1. GA program with high selection pressure
2. SA (Metropolis) program with tabu list and restarting mechanism
3. GA program with high mutation rate
4. GA program with balanced mutation rate and selection pressure
5. GA+GA program – new-strategy program employing two balanced GAs

This demonstration highlights our main results. The SA program with the added tabu list and the restarting mechanism tends to take a lot of time compared to the other programs. GA programs do well except if the selection pressure is high. Under high selection pressure the population quickly converges to the solution, but usually to a wrong one. This program can thus be used as a local optimizer. The high mutation rate GA program tends to find the right solution in reasonable time, but some runs do not converge due to considerable randomness in the population. It is induced by the random changes in the solution parameters in order to bring variety to the population. The best option for the GA is to keep it balanced — in that case most runs get finished in favorable time.

Our strategy of dividing the program into levels is clearly a successful one, since it is much faster than any other we tested. The balanced GA+GA combination gives best performance, while other combinations display lower performance, mainly due to lower efficiency of the SA algorithm or the other GA realizations, respectively.

Application to Data

We have applied the programs to various artificial datasets, containing up to four planets. The time required for a dataset to be fully analysed grows very fast with the number of planets. An improvement in this matter was made by introduction of special single-, double- and triple-planet subroutines replacing the ones for an arbitrary number of planets. However, the computation time was still incomparable to the fast computation of a periodogram. If τ is an average time needed to optimize the parameters of one planet, the sequential approach takes $n\tau$ for n planets, while the simultaneous fit takes approximately τ^n.

The programs were also tested on real data. We used published datasets for UMa 47 [3] and υ And [1] to see if we would confirm the previously published results. As they were confirmed with only minor differences, we have also tried to find a new planet in a data set which has strangely large errors. We used a dataset of radial velocities of HD 89744 from [6] to find several two-planet solutions. Unfortunately, none of the proposed solutions had sufficient significance to be confirmed. This system, which has a massive planet in an eccentric orbit (and thus a well-defined radial velocity shape), demonstrated the importance of

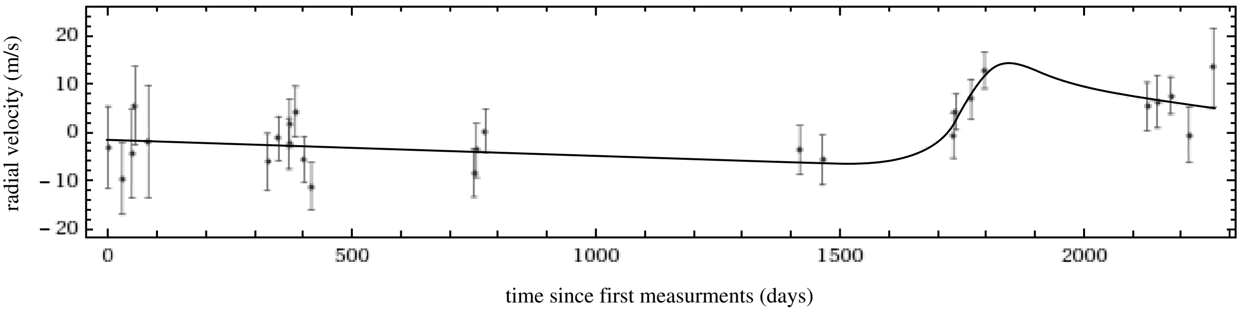

Figure 2: A plot of a possible but not significant signal of a planet orbiting GJ118.

narrowing the parameter space through periodogram analysis. There is no need to search improbable areas of the parameter space if the period and the eccentricity of a planet can be constrained by the periodogram.

We have also applied the programs to our unpublished data from the ESO VLT+UVES survey of red dwarf stars. These datasets did not reveal any significant periodicity. Large errors compared to small amplitudes of signals make many orbital solutions possible, but none of them was found to be significant using bootstrap techniques. An example is given in Figure 2 — a plot of an eccentric planet in 3400 days orbit around star GJ118. Longer observation time would be needed to confirm if this signal is real.

Conclusion

Application to real data has shown that the algorithms are capable of reproducing results previously found by other researchers. However, this approach uses considerably more computer time than the standard sequential Keplerian fit approach, making it less preferable to use as a primary tool for analysis. Although in theory the simultaneous multi-planet fit is the only right way to find the full orbital solution, it may often be impractical to use right from the start. Better performance may be achieved with a suitable combination of this approach and the information extracted from the periodogram, in which case the parameter space is narrowed and made easier to examine. This way the heuristic optimizer is directed towards the regions of the parameter space which are likely to hold the right solution and wasting computer time on the non-prospective regions is thus avoided. Combined with the search strategy introduced in this paper, the resulting program could reach a good efficiency ratio of solution quality to computer time.

Acknowledgements

This work was made possible by the Željka Matutinović Memorial Grant awarded by Almae Matris Alumni Croaticae Mid-Atlantic (AMAC-MA), Students Projects Fund and the grant from the Max Planck Institute for Astronomy.

References

[1] Butler R.P., Marcy G.W., Fischer D. A., et al. ApJ, V. 526, I.2, pp. 916-927 (1999)

[2] Charbonneau, P. ApJS, V. 101, p. 309 (1995)

[3] Fischer D.A., Marcy G.W., Butler R.P., Laughlin G., Vogt S.S. ApJ, V. 564, I. 2, pp. 1028-1034 (2002)

[4] Fischer D.A., Marcy G.W., Butler R.P., et al. ApJ, V. 675, I. 1, pp. 790-801 (2008)

[5] Kirkpatrick S., Gelatt C.D., Vecchi M.P. Science, V. 220, p. 671-680 (1983)

[6] Korzennik S.G., Brown T.M., Fischer D.A., Nisenson P., Noyes R.W. ApJ, V. 533, I. 2, pp. L147-L150 (2000)

[7] Lomb N.R. Ap&SS, V. 39, pp. 447-462 (1976)

[8] Scargle J.D. ApJ, V. 263, pp. 835-853 (1982)