ASTRONOMY & ASTROPHYSICS
SUPPLEMENT SERIES

Astron. Astrophys. Suppl. Ser. **132**, 417–429 (1998)

Determination of orbital parameters of interacting galaxies using a genetic algorithm

Description of the method and application to artificial data

M. Wahde

NORDITA, Blegdamsvej 17, 2100 Copenhagen, Denmark

Received October 31, 1997; accepted April 24, 1998

Abstract. A method for determining the orbital parameters of interacting pairs of galaxies is presented and evaluated using artificial data. The method consists of a genetic algorithm which can search efficiently through the very large space of possible orbits. It is found that orbital parameters close to the actual orbital parameters of the pair can in general be found, even in the presence of fairly high levels of noise in the data. Both position and velocity data can be employed, but if no velocity data are available, the orbital parameters can be determined using position data only. The inner regions of the galaxies, which are difficult to model, can be neglected, and the orbital parameters can be determined using the remaining information.

Key words: galaxies: interactions — galaxies: kinematics and dynamics — methods: N-body simulations

1. Introduction

Observations of interacting galaxies provide important information about galactic structure and evolution. Due to the long time scales (compared to the lifetime of a human) involved in interactions between galaxies, one can only obtain a single snapshot of an interacting system. Dynamical modelling of such systems therefore constitutes an important complement to observations, and can naturally be divided into two parts: Modelling of the individual galaxies participating in an interaction, and determination of the orbital parameters. Advances in the techniques for modelling of individual galaxies in interacting systems has led to a greatly increased understanding of features such as bars, rings, and spiral arms.

In order to understand the dynamics of an interacting system of galaxies it is, however, equally important to know the parameters of the relative orbit of the two galaxies. Determination of orbital parameters has only been carried out for a small fraction of all observed interacting galaxies. An example of a system that has been much studied is the M 51 system (for recent results, see Engström & Athanassoula 1991; Howard & Byrd 1990, and Hernquist 1990). The fact that several authors have arrived at very different orbits shows the difficulties involved in modelling even a well-observed system such as M 51.

Some of the main problems encountered when numerical simulations are used for determining orbital parameters are that the parameter space that needs to be searched is often very large, and that the results of each simulation must be compared with the observational data. While methods for automatic comparison of data between observations and simulations have been used in some cases (e.g. Engström & Athanassoula 1991), very little has been done to find an efficient method for reducing the amount of searching necessary in order to find the orbit in a general case. In this paper, an efficient search method will be presented and evaluated using artificial data.

Science often benefits from the sharing of information between different disciplines. One example of this is the invention of *genetic algorithms* (Holland 1975). With natural evolution as the inspiration, genetic algorithms (hereafter GAs) use artificial selection and the genetic crossover and mutation operators to manipulate strings of numbers which encode the variables of the problem, thereby reaching better and better solutions to the problem. GAs are used in many branches of science (see the Appendix). However, in astrophysics there have, as yet, only been a few applications of GAs, in the fields of solar coronal modelling (Gibson & Charbonneau 1996), helioseismology (Tomczyk et al. 1995), pulsar planet searching (Lazio 1997), eclipsing binary stars (Hakala 1995), and gamma-ray astronomy (Lang 1995). For an excellent review of GAs in astronomy and astrophysics, see Charbonneau (1995).

In this paper, a GA will be used for searching the space of possible orbital parameters for pairs of interacting galaxies. Section 2 contains a description of the problem,

and the method of solution is presented in Sect. 3. The results are given in Sects. 4 and 5. The method is discussed in Sect. 6, and the conclusions are presented in Sect. 7. The Appendix contains a brief description of the essential features of the GA used in the paper, as well as some references for further reading.

2. Description of the problem

The problem to be solved is the following: Given (photometric) observations of a pair of interacting galaxies, as well as systemic velocities for the two galaxies, determine the parameters of the orbit.

In principle, three observations at different times suffice to deduce the orbital parameters of a comet or an asteroid. For galaxies, however, one can only obtain observations of a single snapshot. Fortunately, such snapshots contain a wealth of information since the gravitational forces between the galaxies produce deformations in the form of, for example, arms, tails, and bridges connecting the galaxies. In addition to the position data, the radial velocity field can also be measured. From the complete set of data, information about scale radii, scale heights, disc inclinations, velocity dispersions, and masses can be obtained.

However, positions along the line of sight and velocities in the plane of the sky cannot be measured, and the problem of finding the orbital parameters is therefore far from trivial. Additional complications appear since observations never provide perfect, noise-free data.

A snapshot from a simulation of two face-on galaxies is shown in the left panel of Fig. 1. Clearly, the galaxies are strongly interacting with material being torn off from both. Just looking at the snapshot, it is difficult to say very much about the orbit. Given systemic velocities, positional information of the type shown in Fig. 1, and, in some cases, the radial velocity field, I shall in Sects. 4 and 5 use a GA to determine the orbital parameters of several (artificial) interacting systems.

The purpose of the paper is to test GAs as a method for finding orbital parameters of interacting galaxies, rather than testing the methods used for the individual simulations. In order to study the more violent types of interactions (e.g. mergers) where even the inner regions of the galaxies are strongly deformed, fully self–gravitating simulations, in which the mass distributions of the two galaxies are taken as unknown variables, would be required.

However, in this paper, the discussion will be restricted to less violent types of interactions, for which some simplifications could be made:

First, the scale length ($r_{\rm d}$), inclination, (i) and major axis position angle (PA) of each disc were assumed to be known from measurements of the central regions of the galaxies.

Second, the simulations were non–self–gravitating, i.e. interparticle forces were neglected, the disc particles were influenced only by the gravitational forces from the two point particles (of mass m_1 and m_2), representing each galactic disc, and no dispersion velocities were added. This was the case both for the simulations carried out in order to determine orbital parameters, and for the simulations which generated the artificial observational data.

For the former, it would have been very difficult to use self–gravitating simulations, due to the large number of simulations which had to be carried out for each determination of orbital parameters.

For the latter, self–gravitating simulations could, in principle, have been employed, but it would not have been useful to do so, since the use of self–gravity would introduce a number of complications having, in general, little to do with the performance of the GA. For instance, in a self–gravitating simulation, velocity dispersions must be added and it may also be necessary to add a stabilizing halo, thus introducing several additional model parameters. Also, if a grid code were to be employed, the additional softening caused by the finite size of the grid cells would complicate the comparison even more.

The use of non–self–gravitating simulations for generating the artificial data does, however, somewhat simplify the task of the GA. In such cases, it is possible for the GA to arrive at orbital parameters such that the artificial data are reproduced *exactly*. This, of course, would neither hold for real data nor for data generated by a simulation incorporating self–gravity, where instead the results corresponding to the best orbital parameters found by the GA would only approximate the observational data, rather than reproducing them exactly.

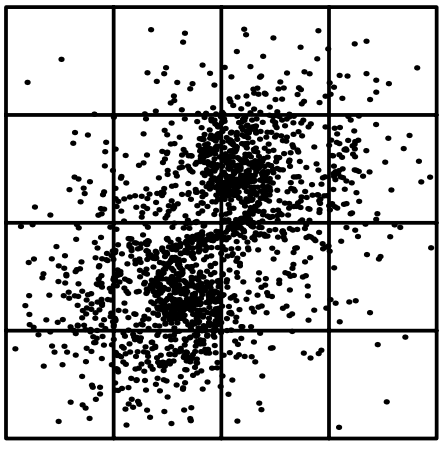
The neglect of self–gravity may not be a serious limitation in the cases where only the outer parts of the two galaxies are significantly affected by the interaction: In a tenuous, low mass arm or tail consisting of material from the outer regions of either galaxy, the self–gravity is usually less important than the tidal field of the two galaxies. The discussion below will deal with such systems, and more violent forms of interactions will *not* be considered.

3. The method

In this section, the method for determining orbital parameters will be presented. In Sects. 3.1 and 3.2, the individual simulations are described. In Sect. 3.3 I describe the GA which was used for searching the space of orbital parameters. A general description of GAs is given in the Appendix, which also contains references to more complete discussions of the subject of GAs.

3.1. The simulations

I have used a coordinate system such that the $x-y$-plane coincides with the plane of the sky (with the $x-$axis



0.003	0.023	0.050	0.014
0.023	0.216	0.465	0.066
0.100	0.580	0.157	0.020
0.036	0.174	0.026	0.004

Fig. 1. Computation of the data for an observation of a pair of interacting galaxies. The grid is superposed on the image of the interacting galaxies, and the grid cells are assigned values corresponding to the mass in each cell. For clarity, only a very coarse grid has been used in this figure

horizontal), and the z-axis pointing *towards* the observer. The longitude of the ascending node (Ω) is measured from the $x-$axis. The orbital inclination is defined to be zero if the orbit lies in the plane of the sky. As mentioned above, I have used non–self–gravitating simulations, in which the galaxies are represented by point masses surrounded by discs of particles which, in turn, are only influenced by the gravity of the two point masses.

As unknown variables I have taken the relative (systemic) velocities between the galaxies in the plane of the sky (Δv_x and Δv_y), the separation along the line of sight (Δz), the masses (m_1 and m_2), and the spins (s_1 and s_2) of the two galaxies. The encoding of the unknown parameters in the strings used by the GA is described in Sect. 3.3.

Observations provide values of the relative systemic velocities along the line of sight (Δv_z) and the separations in the plane of the sky (Δx and Δy).

For each simulation, values of the unknown variables are obtained as described in Sect. 3.3 below, and the orbital parameters can then be determined from the masses (m_1 and m_2), the three components of the separation vector (Δx, Δy, and Δz), and the three components of the relative velocity (Δv_x, Δv_y, and Δv_z). The orbital parameters are a (semi-major axis), e (eccentricity), i (inclination), Ω (longitude of the ascending node), ω (the argument of pericentron), and T (the time of the pericentre passage). The transformations from cartesian coordinates and velocities to the orbital parameters are straightforward and well known, and will not be given here. For a discussion of the transformations involved, see e.g. Danby (1988) or Deutsch (1963).

The orbital parameters are of course not absolutely necessary for orbit integration - the cartesian coordinates and velocities would suffice - but the orbital parameters have the advantage of giving information which is easier to visualize. For instance, the value of e immediately shows whether an orbit is elliptical or hyperbolic.

When the orbital parameters have been found, i.e. decoded from the chromosome (see Sect. 3.3), the positions of the two galaxies are integrated backwards in time. During the backwards integration, the galaxies are represented by point particles moving on a two-body orbit. Starting at time t_0 the backward integration lasts for $N_{\rm step}$ steps. At each step j, the new time is computed as $t = t_0 - j{\rm d}t$, where ${\rm d}t$ is the length of the time step, and Kepler's equation is then solved to yield the eccentric anomaly E (for elliptic orbits) or the 'hyperbolic anomaly'F (for hyperbolic orbits). The case $e = 1$ exactly, i.e. parabolic orbits, is neglected. However, e can be arbitrarily close to 1.

I use sign conventions such that $a < 0$ for hyperbolic orbits. Using the orbital parameters, the relative positions and velocities of the two galaxies can be obtained, through the transformations discussed in e.g. Danby (1988) or Deutsch (1963).

At the end of the backward integration, a disc of particles is added to each of the galaxies. Before the first simulation is carried out, the program reads two copies of a standard disc of unit mass and unit scale length. However, the masses and scale lengths of the two galaxies are generally different from unity, and therefore the positions and velocities of the particles in each disc are, for each simulation, scaled to appropriate values. The directions of

rotation of the two discs are determined by the values of the spin parameters, s_1 and s_2, as discussed in Sect. 3.3 below.

Then, the orbit is integrated forward in time until the final step (corresponding to the time of the observation) is reached, at which point the position and velocity data are stored in the manner described below, and the next simulation can begin.

For hyperbolic orbits the backward integration is straightforward and can be terminated when the galaxies are at sufficient distance (a few galactic radii, say) from each other. The situation for elliptical orbits is more difficult: If the duration of the backward integration is badly chosen, the galaxies may not be sufficiently separated at the start of the forward integration. To avoid this problem, the value of $N_{\rm step}$ can be chosen individually for each orbit in such a way that the backward integration proceeds until the apocentre is reached. Furthermore, the problems encountered for elliptical orbits are more general than that, since previous encounters may have damaged the discs of the galaxies, especially for short period orbits. However, it would be unfortunate to completely neglect elliptical orbits, and thus in Sect. 5 some cases of interacting galaxies on such orbits will be considered as well.

3.2. Evaluation of the simulations

When the final step has been reached, the output from the simulation should be compared with the (artificial, in this paper) observational data. The data from an observation can be either in the form of a contour map, or in the form of a grid of grayscale pixels, the shading at each pixel determined by the amount of light at that point. The version of the method described in this paper requires the data to be of the latter form, even though the program could be generalized to operate on contour maps as well.

Since I only use artificial observations here, I will use as data the amount of mass at each pixel rather than the amount of light. When real data is used, a scale factor (the mass-to-light ratio) must be introduced.

Thus, at the end of each run a grid is superposed on the two galaxies, and the amount of mass in each grid cell is stored, each particle contributing $m_1/N_{\rm p,1}$ if it belongs to the first galaxy or $m_2/N_{\rm p,2}$ if it belongs to the second, where $N_{\rm p,1}$ and $N_{\rm p,2}$ are the number of particles used for the first and the second galaxy, respectively.

The data from the orbit integration which provides the (artificial) observation is stored in the same way, and is read by the program before the first simulation is carried out.

The number of grid points in the $x-$ and $y-$directions, denoted n_x and n_y, as well as the size, $L \times L$, of the (quadratic) grid cells are input parameters to the program, and should obviously be chosen in such a way that the relevant parts of the two galaxies are contained within the grid. For the data presented in Fig. 1, the $x-$ and $y-$coordinates ranged from -14.0 to 14.0 in program units, so if the grid parameters are taken to be, for example, $n_x = n_y = 4$ and $L = 7.0$, the corresponding data will be the matrix of mass values in the right of Fig. 1.

In order to evaluate a given simulation, the deviation between its results and the observational data should be measured. The deviation measure can be defined in different ways, and it will here be defined as

$$\delta = \frac{n_{\rm typ}^2}{n_x n_y} \sum_{i,j} |m_{i,j} - m_{i,j}^{\rm obs}|/(m_\epsilon + m_{i,j}^{\rm obs}), \tag{1}$$

where $m_{i,j}$ is the mass in cell (i, j) obtained from the simulation, $m_{i,j}^{\rm obs}$ is the same quantity obtained from the observational data, and the sum extends over the whole grid. The m_ϵ in the denominator is needed to prevent a divergence in the cases where $m_{i,j}^{\rm obs}$ is zero, and its value is taken to be the typical mass of a particle in a galaxy of unit mass, i.e. $1/N_{\rm p}$ where $N_{\rm p}$ is the number of particles used in the simulations. The factor in front of the summation sign is a normalization factor. Its sole purpose is to make δ independent of the number of grid points used in the data comparison. Thus, $n_{\rm typ}^2$ is a typical value of the number of pixels, here taken to be 49, and $n_x n_y$ is the number of pixels used in the computer run. In the runs discussed in Sects. 4 to 6, $n_x = n_y = 7$, yielding a normalization factor equal to 1. This choice is somewhat arbitrary. However, the values of n_x and n_y must neither be too small nor too large since, in the former case, the tidal features used by the GA will not be resolved and, in the latter case, the data from the observation will be unnecessarily detailed and thereby more sensitive to the noise which is always present in real data.

When real data is used, it is the amount of light at each grid point, rather than the mass, that is detected, and the deviation measure could instead be defined as, for example,

$$\delta \propto \sum_{i,j} |g_{i,j} - g_{i,j}^{\rm obs}|/(\nu + g_{i,j}^{\rm obs}), \tag{2}$$

where g denotes the shading of pixel (i, j). If the g values were normalized to lie between 0 (black, no light) and 1 (white, maximum light), the value of ν could be, say, 0.001. The deviation measures just defined punish strongly those simulations which try to put many particles in regions which are only supposed to contain a few.

Note that, with the deviation measure defined in Eqs. (1) and (2), the GA *does not* make use of the radial velocity field of the two interacting galaxies: The only velocity information used are the systemic velocities of the two galaxies. It is important that the algorithm should be able to function without knowledge of the complete radial velocity field of the interacting system since, in many cases, one does not have access to both accurate position *and* velocity data. However, if the velocity information is available, it should of course be used as it may help the GA

to distinguish between orbits for which the position data appear similar. Thus, in analogy with Eq. (1), a velocity deviation measure can be defined as

$$\delta_v = \frac{n_{\rm typ}^2}{n_x n_y} \sum_{i,j} |\overline{v}_{i,j} - \overline{v}_{i,j}^{\rm obs}| / (|\overline{v}_{i,j}^{\rm obs}| + \overline{v}_\epsilon), \quad (3)$$

where $\overline{v}_{i,j}$ and $\overline{v}_{i,j}^{\rm obs}$ denote the average radial velocities in cell (i,j) obtained from the simulation and from observational data, respectively. The actual value of $\overline{v}_\epsilon$, which serves the same purpose as m_ϵ in Eq. (1), is not of great importance as long as it is not too large, and it can be taken equal to the systemic velocity of the larger of the two galaxies. If the systemic velocity is very large, a typical rotation velocity of either of the galaxies can be used instead.

3.3. The genetic algorithm

In the preceding subsections, the individual simulations were described. In this subsection we shall see how the GA operates. At the start of each GA run, the chromosomes of the $N_{\rm pop}$ simulations (or *individuals* in the biological terminology used in the Appendix) of the first generation are initialized by assigning random numbers between 0 and 9 to each of the genes. The encoding of the variables is illustrated in Fig. 2. As an example of the decoding, the string at the bottom in Fig. 2 would, when decoded, give the values $m_1 = 1.21$, $m_2 = 0.85$, $\Delta z = -14.6$, $\Delta v_x = 0.119$, $\Delta v_y = -0.133$, $s_1 = 1$ (counterclockwise), and $s_2 = -1$ (clockwise). Note that the encoding is decimal rather than binary.

When the chromosomes have been initialized, the program loops through all of the $N_{\rm pop}$ individuals in the first generation. For each individual, the chromosome is decoded and the orbital parameters are computed as outlined in Sect. 3.1. Then, the two orbit integrations (backward and forward) are carried out, the result is compared with the observational data in the manner described in Sect. 3.2, and the resulting value of the deviation is computed and stored. In cases where only position data is available, the function

$$f = \frac{1}{1+\delta}. \quad (4)$$

is used as a fitness measure. The constant in the denominator is introduced to prevent a divergence in the (unlikely) case of δ being equal to zero. If both position and velocity data are available, the fitness measure is instead defined as

$$f = \frac{1}{\sqrt{(1+\delta)(1+\delta_v)}}. \quad (5)$$

Thus, with the fitness functions just defined, $f = 1$ corresponds to a perfect match. When all individuals of the first generation have been evaluated, the fitnesses are ranked using linear fitness ranking as described in the Appendix. Thus, as an example, for the fitness measure defined in Eq. (4), the exact form of the mapping from δ to f will be of no importance when new generations are formed, and any fitness function for which the fitness increases with decreasing δ could have been used. However, as discussed in Sect. 4.2 below, the fitness values play an important role when the quality (i.e. the goodness of fit) of a set of orbital parameters is evaluated.

In order to produce the genetic material, i.e. the chromosomes, of the second generation, the chromosome of the best individual in the first generation is copied twice, and the remaining $N_{\rm pop} - 2$ individuals of the second generation are formed through the process of parent selection followed by crossover and finally mutation as described in the Appendix. When all $N_{\rm pop}$ new individuals have been generated, the genetic material of the first generation is deleted and the evaluation of the second generation can begin. After the second generation has been evaluated, the third generation is formed, all its constituent individuals are evaluated, etc.

In this process, individuals with high fitness values will have a greater chance of spreading their genetic material to the next generation than individuals with low fitness values. To those unfamiliar with GAs, this process may not, at a first glance, seem to be very efficient. However, as many authors have concluded (e.g. Charbonneau 1995) and as we shall see in Sects. 4 and 5, GAs are in fact *very* efficient for solving optimization problems for which the search spaces are large.

3.4. Notes on the simulations

Some general facts about the simulations were given in Sect. 2. In addition to these, the following should also be noted: The disc scale height (which is of little importance for the face-on simulations, but more important for the simulations described in Sect. 5) was set equal to $r_{\rm d}/10$, but any value of the scale height could of course have be employed. In general, the discs consisted of 1 000 particles each for a total of 2 000 particles per simulation. The exceptions are Run 4 (see below) for which a total of 10 000 particles were used, and Runs 6 and 8 for which a total of 4 000 particles were used.

A typical 100 generation computer run with 1 000 particles per galaxy and $N_{\rm pop} = 500$ requires 12.3 CPU hours on a Sun Enterprise 2.

The units used are such that G, the gravitational constant, is equal to 1. The unit of length can be taken to be 1 kpc, the unit of time 1.05 Myr, and the unit of mass $2\,10^{11} M_\odot$. The unit of velocity is then equal to 931 km s^{-1}. Other scalings to physical units can, of course, be used as well. In the genetic encoding scheme, the galaxy masses range from 0.01 to 99.99, the separation along the line of sight (Δz) ranges from -999.9 to 999.9, and the relative velocities (Δv_x and Δv_y) range from -9.999 to 9.999. The spins can take the values -1 and 1. The total number of combinations is 3.2×10^{21} (!).

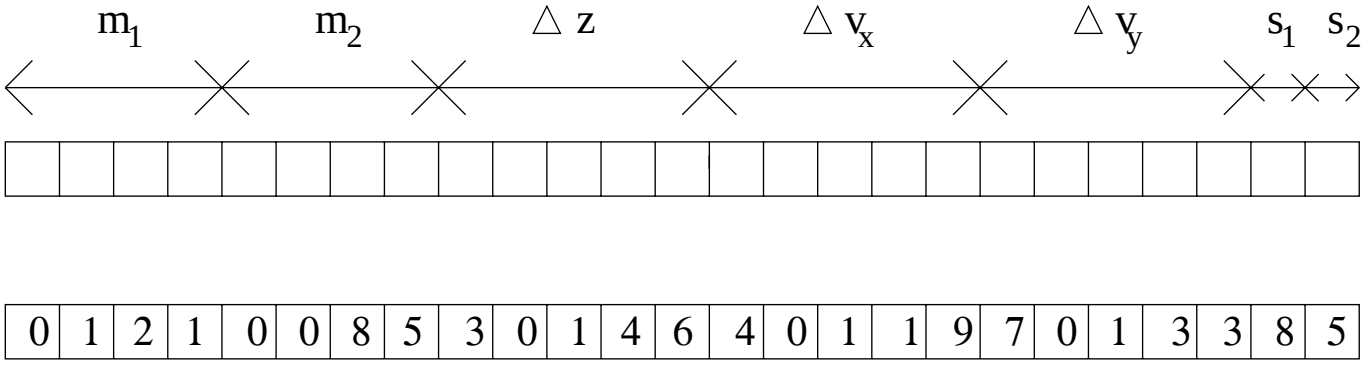


Fig. 2. Top: The encoding scheme for the seven unknown variables. Genes 1-4 encode the mass of the first galaxy such that $m_1 = g_1 * 10^1 + g_2 * 10^0 + g_3 * 10^{-1} + g_4 * 10^{-2}$, where g_i denotes the value of gene i. The other variables are encoded in a similar fashion. Genes 9, 14, and 19 encode the signs of Δz, Δv_x, and Δv_y, respectively, such that the sign is negative if the corresponding value is odd and positive if it is even. The spin of the first galaxy is -1 (clockwise) if the value of gene 24 is odd, and $+1$ if it is even. The spin of the second galaxy is encoded, in the same way, in gene 25. Bottom: an example of a chromosome. Decoding the chromosome, the values $m_1 = 1.21$, $m_2 = 0.85$ etc. are obtained

Often, however, the full ranges just described need not be used. For instance, if two galaxies are strongly interacting and connected by a bridge, one does not expect them to be almost a thousand kpc apart along the line of sight! If, for instance, the values of Δz can be restricted to the interval -50 to 50, say, the number of possible values of Δz is reduced from 20 000 to 1 000. Even with such reductions, the search space will still be very large, and an efficient search method is therefore needed.

In all runs, the crossover rate (see Appendix) was equal to 1.0, and elitism was used.

4. Results from simple applications of the method

4.1. Run 1

In this section, the results of applying the method presented above will be given. Applications to systems with complicated interaction geometries will be deferred to Sect. 5. Thus, this section will serve as an introduction to the basic features of the method, and the discussion will be limited to simple face-on systems for which only position data are known.

First, the system shown in the left panel of Fig. 1 will be studied: In order to determine the corresponding orbital parameters, a run with population size $N_{\rm pop} = 500$ was carried out. The number of generations ($N_{\rm gen}$) was 100, the mutation rate was $p_{\rm mut} = 0.003$, and the number of grid cells was 49 ($n_x = 7, n_y = 7$). For Run 1 and all other runs described in this section, the values of m_1 and m_2 were constrained to lie between 0.3 and 3.0, Δz between -50.0 and 50.0, and Δv_x and Δv_y between -0.999 and 0.999. All possible spin combinations were allowed. The total number of combinations of the unknown variables $m_1, m_2, \Delta z, \Delta v_x, \Delta v_y, s_1, s_2$ was then approximately $1.2\ 10^{15}$. Mutated chromosomes were only accepted if their values of the unknown parameters were contained in the above intervals. For the observational data used in Run 1, the actual values of m_1, m_2, Δz, Δv_x, Δv_y, s_1, and s_2 were 1.0, 1.0, 3.0, -0.672, 0.839, 1, and 1, respectively. Table 1 presents the results of Run 1: The first 6 rows show the orbital parameters of the best simulation in generations 1, 5, 10, 20, 50, and 100, and the final row shows the actual orbital parameters of the "observed" system. As can be seen, the GA was able to find the orbital parameters with great accuracy. In fact, acceptable orbital parameters were obtained already after 20 generations. After 100 generations, the image of the data (left panel of Fig. 1) and the image obtained from the best simulation were almost identical. This is partly due to the fact that the same disc distributions were used both for the run and for generating the observation. However, the method does not require perfect data of the kind used in this run and it can, in fact, cope with quite high levels of noise, as discussed in Sect. 6.3 below.

4.2. Additional runs

The results of three additional runs are shown in Table 2. The upper row in each pair shows the orbital parameters of the best simulation in the final generation, and the lower row shows the actual orbital parameters of the (artificial) observation. The grids used for data comparison were again of size 7×7, and the population size was equal to 500 for all runs. 1 000 particles per galaxy were used in Runs 2 and 3. In Run 4, 5 000 particles per galaxy

Table 1. Results from Run 1. The first 6 rows show, for the best simulation of generations 1, 5, 10, 20, 50, and 100, the orbital parameters a, e, i, ω, Ω, and F_0, as well as the spins, s_1 and s_2, and the masses, m_1 and m_2, of the galaxies. The parameter F_0 is the hyperbolic anomaly at the final time step of each simulation, from which T, the time of the pericentre passage, can be computed. The final row shows the actual orbital parameters used for generating the left panel of Fig. 1

Gen.	a	e	i	ω	Ω	F_0	s_1	s_2	m_1	m_2
1	−1.247	11.34	52.85	293.9	121.9	−1.239	−1	1	0.91	1.05
5	−8.722	1.836	37.99	324.4	61.32	24.24	1	1	0.79	0.92
10	−12.49	1.687	45.65	314.7	81.96	12.09	1	1	0.90	1.05
20	−2.105	4.796	24.13	17.70	18.52	27.77	1	1	0.98	1.00
50	−2.097	4.594	23.49	13.48	16.55	33.21	1	1	0.99	1.00
100	−2.231	4.416	23.96	13.01	18.03	32.03	1	1	0.99	1.00
Obs.	−2.184	4.466	23.96	14.33	15.75	33.00	1	1	1.00	1.00

Table 2. Results from Runs 2 to 4. For each pair of rows, the upper row shows the orbital parameters of the best simulation in generation 100, and the lower row shows the orbital parameters corresponding to the observational data. Upper pair: Run 2, middle pair: Run 3, lower pair: Run 4

Gen.	a	e	i	ω	Ω	F_0	s_1	s_2	m_1	m_2
100	−8.886	2.215	47.04	334.3	307.3	33.40	1	1	1.10	0.89
Obs.	−8.193	2.409	44.08	337.3	310.8	27.86	1	1	1.08	1.02
100	−16.26	1.501	70.10	112.9	309.2	28.58	1	1	1.30	0.51
Obs.	−13.83	1.589	67.76	113.5	309.6	30.07	1	1	1.28	0.54
100	−2.196	4.477	24.36	15.90	14.73	32.57	1	1	0.99	1.00
Obs.	−2.184	4.466	23.96	14.33	15.75	33.00	1	1	1.00	1.00

were used. The artificial observation used in Run 4 was generated with the same orbital parameters as those used for generating the observational data of Run 1, to facilitate comparison between the two runs. As is evident from the table, acceptable orbital parameters were found in all cases.

Not all determinations of orbits proceed as smoothly as, for example, Run 1. In order for the GA to be able to find the orbit, it is a requirement that the galaxies show some signs of interactions, i.e. distortions of some form. This was the case for Runs 1 to 4. However, in a case where the observation consisted of two more or less unperturbed discs, the GA was not able to find the orbit. The fact that signs of interactions are needed is rather obvious and does not, in practice, imply any restrictions. After all, the method is intended for *interacting* systems.

A more important problem is that, even for clearly interacting systems, the GA is not always able to find the correct orbit on the first attempt. Since the population size is far from infinite, there may simply never be sufficient variation in the genetic material to obtain the correct orbital parameters, at least if $p_{\rm mut}$ is small. $p_{\rm mut}$ can of course be increased, but the larger its value, the more the GA approaches a random search. Fortunately, it is always easy to distinguish between a run that fails and one that succeeds, namely through the fitness values. In a successful run, the fitness values continue to increase throughout the run, whereas in an unsuccessful run, the GA rather quickly gets stuck at low values.

For the excellent fit obtained for Run 1, the fitness of the best simulation was 0.430. The corresponding numbers for Runs 2 to 4 were 0.192, 0.339, and 0.475[1]. In contrast, in a run that fails, fitness values above 0.1 are not reached, and usually the GA gets stuck at even lower values. Even in such situations improvements do occur, but at a very slow rate, and it is usually faster to restart the run, using different values of the random number generator seed and the mutation rate.

The results for Runs 1,2, and 4 were obtained on the first attempt, but Run 3 required two attempts. In the first attempt, with mutation rate $p_{\rm mut} = 0.003$, the GA got stuck at a suboptimal solution with fitness 0.0863. The mutation rate was then increased to 0.010 and the random number generator seed was changed, resulting in a maximum fitness of 0.339 in the second attempt.

[1] The fitness increases steeply when the final details of the fitting are carried out, and therefore fitness values above 0.5 are rarely reached.

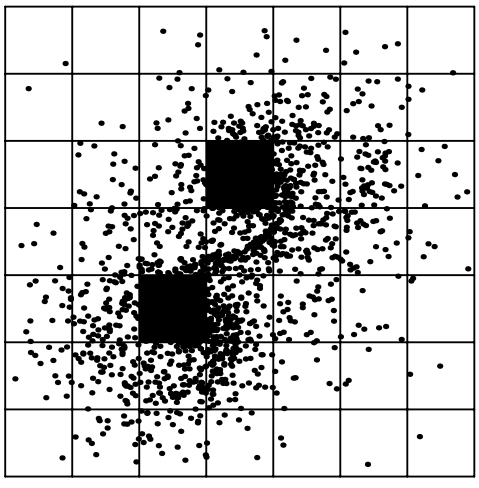


Fig. 3. The two regions of highest density were discarded, as indicated by the black squares. The original picture, before discarding the data in the two black squares, was identical to the left panel of Fig. 1

4.3. Blocking out the inner regions of the galaxies

As mentioned previously, tidal features such as bridges and tails are used by the GA when it attempts to determine the orbital parameters of an observed system. However, unlike the artificial data used here, real observational data is made complicated by the existence of features such as bars, rings, ovals, etc. in the inner regions of the galaxies. Furthermore, in an observation for which the tidal features are clearly seen, the inner regions may be saturated, i.e. overexposed. Thus, in order to avoid problems caused by the appearance of the inner regions, a modified version of the GA program was constructed such that any pixel in the grid could be blocked out and discarded. An example of a run with the modified program is illustrated in Fig. 3, where the two pixels with highest density were discarded.

Thus, in this run, the deviation was computed using only the 47 (i.e. $n_x \times n_y - 2$) remaining pixels. Clearly, this problem is more difficult to solve for the GA (or anyone else!), since only partial information is accessible. The result of the run is shown in Table 3, the upper row as usual showing the orbital parameters of the best simulation and the lower row showing the observational data, which were the same as for Run 1. Even though the resulting fit is not as stunning as for Run 1, acceptable orbital parameters were obtained.

5. Applications of the GA to more complicated interaction geometries

While the runs discussed in the previous section served to illustrate the basics of the algorithm, they were somewhat oversimplified compared to the cases encountered when actual interacting systems are studied. Thus, in this section, some more stringent tests of the algorithm will be presented.

Fig. 4. The observations corresponding to Run 6 (upper panel) and Run 8 (lower panel)

First, it is rare that observed interacting galaxies are exactly face-on, and that assumption will now be abandoned by allowing arbitrary disc plane inclinations and major axis position angles for both discs. Thus, four new parameters, hereafter denoted i_1, i_2, PA_1, and PA_2, need to be introduced. Now, for those interacting systems for which the method described here is best adapted, i.e. systems such that the outer parts of the galaxies are affected but the inner parts of the discs are left intact to a great extent, such that the scale lengths and scale heights of the discs can be measured, the parameters i_1, i_2, PA_1, and PA_2 can also normally be measured, albeit with some uncertainties. Hence, these angles will in general be assumed to be measurable and will be made part of the input data. A case in which the inclinations and position angles are instead made part of the set of variables will, however, also be discussed.

Second, the orbits used in Sect. 4 were all hyperbolic. In this section elliptical orbits will be considered as well.

Table 3. Results from the run illustrated in Fig. 3. The upper row shows the orbital parameters of the best simulation in generation 100, and the lower row shows the orbital parameters corresponding to the observational data

Gen.	a	e	i	ω	Ω	F_0	s_1	s_2	m_1	m_2
100	−2.483	4.126	25.99	17.78	13.37	31.69	1	1	0.99	1.00
Obs.	−2.184	4.466	23.96	14.33	15.75	33.00	1	1	1.00	1.00

Table 4. Results from Runs 5 to 8. In Runs 5 and 6, elliptical orbits were used, and in Runs 7 and 8, the orbits were hyperbolic. As shown in the last four columns of the table, at least one of the galaxies had a non-zero inclination in all cases. The information concerning Run 7 contains four rows, since it was rerun twice: once without making use of the velocity information, and once with the inclinations and position angles as variables. The observations corresponding to Runs 6 and 8 are shown in Fig. 4

Gen.	a	e	i	ω	Ω	E_0/F_0	s_1	s_2	m_1	m_2	i_1	PA_1	i_2	PA_2
100	12.55	0.669	118.7	142.4	334.2	44.20	−1	1	1.00	0.19				
Obs.	11.89	0.653	118.9	141.2	333.8	45.69	−1	1	1.00	0.20	20.0	0.0	50.0	0.0
100	10.86	0.635	117.9	57.92	65.18	71.12	1	1	1.02	0.11				
Obs.	10.81	0.698	115.7	57.50	69.47	72.55	1	1	0.90	0.11	60.0	0.0	90.0	−45.0
100	−4.362	1.858	59.36	17.20	273.6	70.58	−1	1	0.99	0.29				
100, no vel. data.	−4.285	1.804	58.30	10.89	279.2	73.79	−1	1	0.89	0.22				
200, incl., P.A.	−4.413	1.890	61.90	16.09	273.0	72.13	−1	1	0.92	0.28	46.0	−43.0	58.0	34.0
Obs.	−4.612	1.825	59.77	16.32	274.2	69.85	−1	1	1.00	0.30	45.0	−45.0	60.0	30.0
100	−4.372	2.503	85.91	113.6	47.02	27.83	1	1	0.91	0.39				
Obs.	−5.299	2.121	84.99	109.8	47.06	31.17	1	1	1.00	0.40	80.0	−40.0	0.0	0.0

The elliptical orbits used will be fairly eccentric ($e > 0.5$) and will be chosen such that the discs are not severely damaged at pericentre passages.

Thus, the simulations to be described below constitute a greater challenge to the genetic algorithm. However, there is also an additional tool yet to be used, namely the radial velocity field. The use of the velocity field may be particularly useful for e.g. highly inclined systems for which the position data may yield ambiguous information.

The results from four runs with inclined discs are shown in Table 4. In these runs, both position and velocity data were used by the GA. As is evident from the table, the algorithm was able to find orbital parameters close to the correct ones in all cases. Note that the two first pairs shown in Table 4 correspond to elliptical orbits.

In order to test the importance of the velocity data, Run 7 was repeated using only position data, and the result from this run is shown in Table 4 as well. Clearly, the use of the velocity data improved the results, even though acceptable orbital parameters were found even in the case in which only position data were used.

For real systems, the inclinations and position angles are rarely known with high accuracy. In order to take this into account, the simulation program was modified to include the four angles i_1, i_2, PA_1, and PA_2 in the set of unknowns. Since the values of these angles can at least be estimated, they were allowed a range of variation of only ± 10 degrees, in steps of 1 degree, thus increasing the size of the search space by a factor $21^4 = 194,481$. Under these conditions, Run 7 was repeated once more and the results, including the inclinations and position angles found by the GA, are shown in the seventh row of Table 4. Due to the large increase in the size of the search space, the GA was allowed to run for 200 generations in this case, and, as can be seen from the table, acceptable orbital parameters were found in this case as well.

The observations corresponding to Runs 6 and 8 are shown in Fig. 4. Of particular interest is the fact that the algorithm is able to cope with the difficult case in which one of the discs is edge-on. Thus, the results reported in this section show that the GA method can successfully be applied to realistic interaction geometries.

6. Discussion

6.1. Comparison with other methods

As mentioned in the introduction, very little has been done to find efficient search methods for the problem of determining orbital parameters in a general case. An exception is the work by Salo & Laurikainen (1993) in which an error minimization technique was used to fit the orbital parameters of NGC 7753/7752. However, in their paper, only three variables were taken as fitting parameters for

the error minimization technique and other variables, e.g. the masses, were kept fixed.

A common and straightforward method for finding orbital parameters is to carry out a full search of (part of) the parameter space. The drawback with such a method is that the number of simulations needed to achieve acceptable resolution over the parameter space grows very fast in an N-dimensional space where $N \geq 5$. A simple example should suffice as illustration: In our Run 1, a very close match was obtained after 100 generations consisting of 500 simulations each, i.e. after 50 000 simulations. A full parameter search would only allow $(50\,000/4)^{1/5} \approx 6.59 \approx 7$ values of each of the parameters $m_1, m_2, \Delta z, \Delta v_x$, and Δv_y. The factor 4 in the denominator comes from the fact that, for each set $m_1, m_2, \Delta z, \Delta v_x, \Delta v_y$, four values of the spin must be tested. Even if the spins were assumed to be known, only $50\,000^{1/5} \approx 8.7 \approx 9$ values of each parameter could be tested. With the range $[-50.0, 50.0]$ for Δz, as used in Run 1, that would give a resolution of only $100/(9-1) = 12.5$ and an equally poor resolution for the the other parameters. Note also that, for Run 1, the GA found an acceptable solution already after 20 generations, or 10 000 simulations. With 10 000 simulations available, a full search would allow only $6.3 \approx 6$ values of each parameter to be tested, if the spins were known.

Clearly, a complete, unbiased search of the whole parameter space is not an option, at least when the search space is of dimension five or higher. Thus, in order to carry out a full search, either the number of dimensions must be reduced by fixing the values of some parameters, or, if all parameters are used, some constraints must be imposed on the values to be searched. While this is certainly possible in some cases, the constraints may not always be justified and it may be impossible to find the correct orbit after imposing the constraints. A strong advantage of the GA method is that only very loose constraints need to be imposed, even when the number of parameters is large.

6.2. Comparison with random search

Even though mutations (which play a subordinate role) are random, *selection* is not, and a GA is strongly different from a random search. In order to illustrate the non-randomness of the GA method, a calculation was carried out in which the values of the parameters $m_1, m_2, \Delta z, \Delta v_x, \Delta v_y, s_1$, and s_2 were generated at random. A total of 10 000 random sets of parameters were generated, corresponding to 20 generations of 500 simulations each for the GA. In Fig. 5, the results of the first 20 generations of Run 1 are compared with the results of the random run. The GA took the lead already after 6 generations, and the difference in performance should be obvious from the figure.

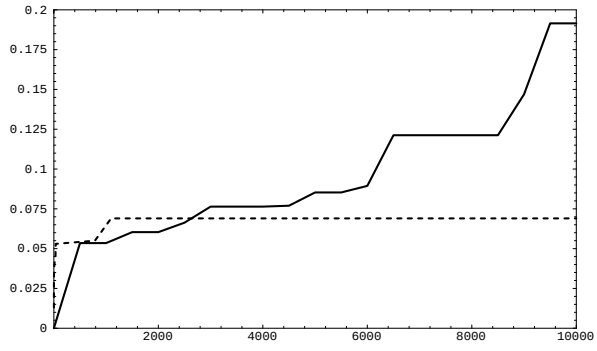


Fig. 5. Comparison of the performance of a GA and a random search. The dashed line corresponds to the random search. The vertical axis shows fitness values and the horizontal axis the number of individuals evaluated

6.3. Sensitivity to noise

Whereas the artificial data sets used so far were noise-free, real data sets are invariably noisy. Furthermore, even if the noise levels are low, in real data there will always be other deviations from the idealized situation used in the GA simulations. For example, the mass-to-light ratio need not be constant. Therefore, in order to be useful, the GA method must be able to function even in the presence of noise. I have tested the noise sensitivity by first adding 10% noise to the data set used in Run 1. The noise was added by changing the values of the masses in the grid of observational data according to

$$m_{i,j} \rightarrow m_{i,j}(1 + \alpha_{i,j}), \tag{6}$$

where $m_{i,j}$ denotes the mass in cell (i, j), and $\alpha_{i,j}$ are random numbers between $-\alpha_{\max}$ and $\alpha_{\max}$, where $\alpha_{\max}$ is the noise level. Thus, for the run with 10% noise, $\alpha_{\max}$ was equal to 0.1. The results are shown in the upper row of Table 5, from which it is evident that 10% noise does not stop the GA from finding the correct solution. The results from a run with 30% noise are shown in the middle row of Table 5. Even in this case, the GA manages to find a solution fairly close to the correct one.

7. Conclusions and directions for further work

I have presented an efficient method, based on genetic algorithms, for finding the orbital parameters of interacting galaxies, and applied it to simulated galaxies on both hyperbolic and elliptical orbits. Given the centre-of-mass positions of both galaxies, their radial velocities, the scale radii and orientation of their discs, and a matrix of observed (light or mass) density data, in some cases supplemented with velocity data, the GA was able to find the orbital parameters with great accuracy in most cases. In the cases where an orbit could not be found, the failure could be detected rather quickly from the fitness values.

The GA can operate on reduced data sets, in which the inner regions have been blocked out, enhancing the

Table 5. Results from the two runs with noisy input data. The upper row shows the orbital parameters of the best simulation in generation 100 for the run with 10% noise added, the middle row shows the same parameters for the run with 30% noise added, and the bottom row shows the orbital parameters corresponding to the observational data. Note that an exact match is impossible in this case, because of the noise added to the data

Gen.	a	e	i	ω	Ω	F_0	s_1	s_2	m_1	m_2	Remark
100	−2.221	4.398	23.77	12.65	17.43	32.88	1	1	0.99	1.02	10% noise
100	−1.901	5.045	23.33	16.14	16.01	31.79	1	1	0.98	0.95	30% noise
Obs.	−2.184	4.466	23.96	14.33	15.75	33.00	1	1	1.00	1.00	

tidal features which are needed for the determination of the orbital parameters.

Even though there have been several simplifications in the test cases used here, the GA method is a promising approach to the type of problems considered in the paper. In order to further improve the method, the possibility of including the scale radii, scale heights, inclinations, and position angles of the two discs in the chromosomes could also be considered. However, as mentioned above, these parameters can often be measured, and the benefits obtained by including them in the chromosomes may not be sufficiently strong to justify the added complexity thus introduced. With further increases in computer speed, another improvement could come from using a simulation code incorporating the self–gravity of the two galaxies. Dark matter haloes could also be added, and if the distributions of dark matter were chosen in such a way as to be easily parametrized by one or a few parameters, it would be possible to add also these parameters to the set of unknowns.

Still, even in its more primitive present state, the GA method is very useful for finding orbital parameters. The output orbital parameters of the best simulation in a GA run can be used as input parameters for an advanced self–gravitating N-body simulation incorporating gas dynamics and dark matter.

Appendix

The aim of this Appendix is only to introduce and describe some of the most elementary features of GAs. For a much more complete discussion of GAs and their performance compared with other algorithms, see Holland (1975), Davis (1991), or Mitchell (1996). For a review of GAs in astronomy and astrophysics, the article by Charbonneau (1995) is strongly recommended.

GAs have been applied in many different subjects, including machine learning, population genetics, neural network design, economics etc. Among other things, GAs are well suited for search and optimization, and are particularly useful when the search spaces are large.

Since GAs are inspired by natural evolution, the terminology often involves terms from biology, such as genes, populations, fitness etc. For an introduction to the terminology, see e.g. Charbonneau (1995) or Mitchell (1996). Whenever such terms are introduced for the first time in this Appendix, they will be put in *italics*, and hopefully their meaning should be clear from the context.

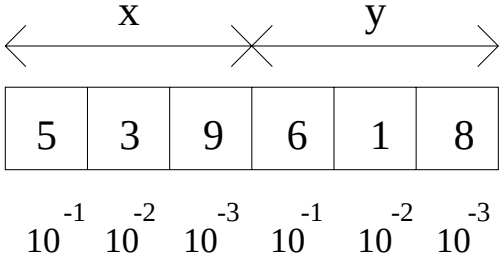


Fig. 6. A typical chromosome of an individual in the inverse-function example. The first gene codes for the first decimal of x, the second gene for the second decimal of x etc. Decoding the chromosome, the values $x = 0.539$ and $y = 0.618$ are obtained

When a GA is to be applied to an optimization problem, the variables of the problem are first encoded in strings (of given length) of integers. Initially, a *population* (i.e. a set) of $N_{\rm pop}$ *individuals* are formed by randomly generating such a string for each individual. Each string constitutes the *chromosome* (i.e. the genetic material) for the individual. The encoding can be either binary or decimal such that the values at the different *genes* (i.e. locations) along the string are integers in the range [0, 1] (for the binary case) or [0, 9] (for the decimal case). The whole set of $N_{\rm pop}$ individuals with their corresponding chromosomes constitutes the first *generation*.

As a trivial example, (the "inverse function example"), imagine that one wishes to find a pair of numbers (x_0, y_0) such that a given function $h(x, y)$ takes a particular value $h_0 = h(x_0, y_0)$. For simplicity, assume that $h(x, y)$ takes the value h_0 only at the one point (x_0, y_0) and that x and y both lie in the interval [0, 1]. If decimal encoding with three digit accuracy is used, the chromosome of an individual could have the form shown in Fig. 6.

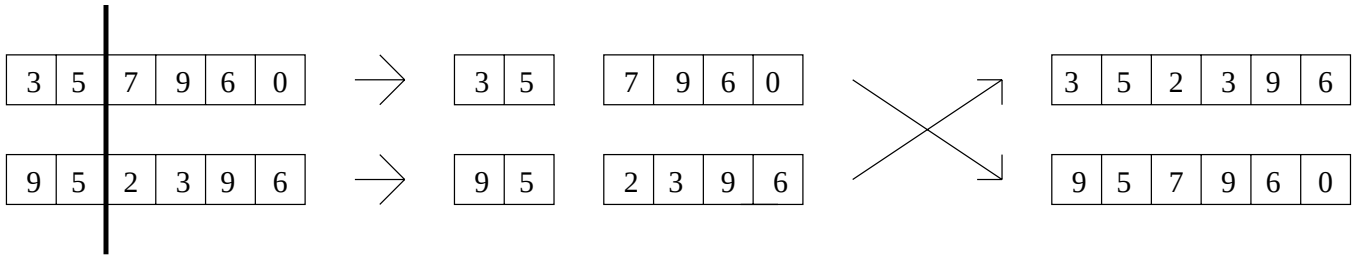


Fig. 7. The crossover procedure. The chromosomes are divided at the crossover point, which is indicated by a thick line, and the parts are joined as shown above

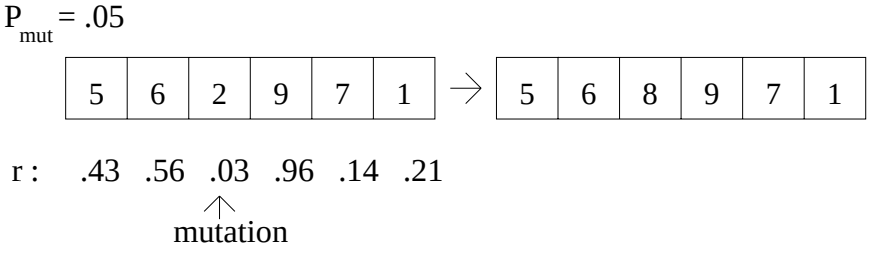


Fig. 8. The mutation procedure. For each of the six locations along the string, a random number r between 0 and 1 is generated and compared with p_{mut}. If r is smaller than p_{mut}, a new, random value is assigned to the gene

When the first generation has been formed, the *fitness* of its constituent individuals should be evaluated. Thus, for each individual, the variables are obtained by decoding the chromosome. Given those variables, the relevant computation can be carried out. In the inverse function example, the computation consists of forming $h(x, y)$ using the values of x and y. Then the result of the computation is compared with the desired result, and a fitness value is assigned such that the smaller the deviation from the desired result, the higher the fitness.

If $h(x, y) = \mathrm{e}^{xy}$ in the inverse function example, and one is looking for values x and y in [0,1] such that $h(x, y) =$ e (the correct solution of course being $x = 1$, $y = 1$), then the individual shown in Fig. 6 would give the value $h(0.539, 0.618) = 1.39529$, the deviation would be $\delta = e - 1.39529$, and the corresponding fitness value f could be defined as $f = 1/(1 + \delta)$.

When all the individuals of the first generation have been evaluated and fitness values have been assigned, the second generation is formed by applying various procedures inspired by natural evolution to the chromosomes of the individuals in the first generation. These procedures include *selection* (followed by *crossover*) and *mutation*.

In order to perform a crossover between two chromosomes, two *parents* are selected from the generation just evaluated. The choice of parents is made in such a way that individuals with higher fitness have a greater probability of being selected than individuals with lower fitness. The fitness values can either be used directly, or some more sophisticated method can be employed. Linear fitness ranking is one example of such a method, in which the individuals are sorted according to their fitness and the best individual is assigned a new fitness equal to N_{pop}, the second best is assigned a new fitness equal to $N_{\mathrm{pop}} - 1$, and so on. This procedure enhances the differences between the individuals, especially if their original fitness values (before ranking) are very similar to each other.

There exist several methods of choosing parents, and here only one of the simplest shall be discussed, namely roulette-wheel selection. When this selection method is used, the sum of the fitnesses $f_i, i = 1, 2, ..., N_{\mathrm{pop}}$ is formed, a random number r between 0 and $\sum_i f_i$ is generated, and the first individual i which satisfies the condition

$$\sum_{j=1}^{i} f_j \geq r, \tag{7}$$

is selected as a parent. As an example, if $N_{\mathrm{pop}} = 3$ and the fitness values are 2, 5, and 3, the first individual is selected if $r \leq 2$, the second is selected if $2 < r \leq 7$ and the third is selected if $r > 7$. When two parents have been chosen (usually with replacement, i.e. such that an individual can be chosen several times), crossover is performed

by dividing the chromosomes of the two parents into two parts, and joining the parts as shown in Fig. 7. The point at which the cut is performed is called the *crossover point*.

When crossover is carried out, two partial solutions to the problem can be joined to form a full solution. Returning to the inverse function example with $h(x, y) = e^{xy}$ as above, it is clear that the two parents in Fig. 7 would both be rather far from the correct solution $x = 1, y = 1$. However, the second of the two new individuals (bottom row in the figure) formed by crossover would be much closer to the solution and would obtain a high fitness value.

Thus, in this way, a new set of chromosomes is formed. Usually not all new chromosomes are formed by crossover. Instead, a *crossover rate* (denoted p_c), is introduced such that crossover is applied to a given pair of parents only if $q \leq p_c$, where q is a random number between 0 and 1. If $q > p_c$ the two parents are copied without modification. Finally, mutation is applied to the new chromosomes. In order to perform mutation on a chromosome, a random number r is generated for each gene along the string, and the condition $r < p_{mut}$, where p_{mut} is the mutation probability, is tested. If the condition is satisfied, the value of the gene is changed to a new random value. The procedure is illustrated in Fig. 8.

Sometimes, the best chromosome(s) are copied directly into the next generation (i.e. without crossover *or* mutation). This is referred to as *elitism*.

The chromosomes thus obtained (or, more strictly, the individuals corresponding to the chromosomes) constitute the second generation. The individuals of the second generation are then evaluated and fitness values are assigned to each individual, after which the third generation is formed etc. This process continues until an acceptable solution to the problem has been found.

The description above only scratches the surface of the vast subject of GAs and the interested reader is again referred to the references cited at the beginning of the Appendix.

Acknowledgements. I would like to thank Dr. K.J. Donner for carefully reading the manuscript, and an anonymous referee for many helpful and constructive comments. I dedicate this work to the memory of my friend and thesis advisor Dr. Björn Sundelius.

References

Charbonneau P., 1995, ApJS 101, 309
Danby J.M.A., 1988, Fundamentals of celestial mechanics. William-Bell, Richmond
Davis L., 1991, Handbook of genetic algorithms. Van Nostrand Reinhold, New York
Deutsch R., 1963, Orbital dynamics of space vehicles. Prentice-Hall, Englewood Cliffs
Engström S., Athanassoula E., 1991, in: Sundelius B. (ed.) Dynamics of disc galaxies. Göteborg University and Chalmers University of Technology, Göteborg, p. 215
Gibson S., Charbonneau P., 1996, BAAS 188, # 36.22
Hakala P.J., 1995, A&A 296, 164
Hernquist L., 1990, in: Wielen R. (ed.), Dynamics and interactions of galaxies. Springer, Berlin, p. 108
Holland J.H., 1975, Adaptation in natural and artificial systems, 1st ed. University of Michigan Press, Ann Arbor; 2nd ed.: 1992. MIT Press, Cambridge
Howard S., Byrd G.G., 1990, AJ 99, 1798
Lang M.J., 1995, Ir. Astron. J. 22, 167
Lazio T.J., 1997, PASP 109, 1068
Mitchell M., 1996, An introduction to genetic algorithms. MIT Press, Cambridge
Salo H., Laurikainen E., 1993, ApJ 410, 586
Tomczyk S., Charbonneau P., Schou J., Thompson M.J., 1995, in: Hoeksema J.T., Domingo V., Fleck B., Battrick B. (eds.) Proc 4th SOHO Workshop: Helioseismology, ESA, Noordwijk, p. 271