Celest Mech Dyn Astr (2009) 105:211–227
DOI 10.1007/s10569-009-9223-4

ORIGINAL ARTICLE

Cooperative evolutionary algorithm for space trajectory optimization

Matteo Rosa Sentinella · Lorenzo Casalino

Received: 14 October 2008 / Revised: 3 June 2009 / Accepted: 13 July 2009 /
Published online: 7 August 2009

Abstract A hybrid evolutionary algorithm which synergistically exploits differential evolution, genetic algorithms and particle swarm optimization, has been developed and applied to spacecraft trajectory optimization. The cooperative procedure runs the three basic algorithms in parallel, while letting the best individuals migrate to the other populations at prescribed intervals. Rendezvous problems and round-trip Earth–Mars missions have been considered. The results show that the hybrid algorithm has better performance compared to the basic algorithms that are employed. In particular, for the rendezvous problem, a 100% efficiency can be obtained both by differential evolution and the genetic algorithm only when particular strategies and parameter settings are adopted. On the other hand, the hybrid algorithm always attains the global optimum, even though nonoptimal strategies and parameter settings are adopted. Also the number of function evaluations, which must be performed to attain the optimum, is reduced when the hybrid algorithm is used. In the case of Earth–Mars missions, the hybrid algorithm is successfully employed to determine mission opportunities in a large search space.

Keywords Trajectory optimization · Evolutionary algorithms · Mission planning · Hybrid algorithm · Earth–Mars round-trip mission

1 Introduction

Evolutionary algorithms (EAs) are optimization procedures which search for the solution that maximizes or minimizes a given function in a prescribed search space. Each solution (individual) is represented by the integer or real values of a finite number of variables, which can vary in prescribed intervals. The optimization procedure is started by producing, usually in a

M. Rosa Sentinella · L. Casalino (✉)
Diparimento di Energetica, Politecnico di Torino, Corso Duca degli Abruzzi, 24, 10129 Torino, Italy
e-mail: lorenzo.casalino@polito.it

M. Rosa Sentinella
e-mail: matteo.rosasentinella@polito.it

random way, an initial population of individuals. Each algorithm is characterized by its own rules that force the evolution of the population in order to favor the improvement of the function to be optimized. Some parameters, typical of each algorithm, control the evolution and determine the capability of finding the optimal solution. Genetic algorithms (GAs), differential evolution (DE), and particle swarm optimization (PSO) are used in the present article.

Initial applications of EAs to space trajectory optimization mainly employed GAs in conjunction with gradient-based methods (Gage et al. 1995; Rauwolf and Coverstone-Carroll 1996, 1997; Hartman et al. 1998). GAs have also been used with calculus of variations (Crain et al. 2000; Woo et al. 2006; Rosa Sentinella and Casalino 2006) for low thrust trajectories; a combination of artificial neural networks with EAs has been applied to solar sail trajectories (Dachwand and Wie 2007). An interesting application of GAs (Conway et al. 2007) concerns their use in hybrid optimization problems (i.e., those problems with both continuous and categorical variables), in conjunction with other techniques.

EAs are better suited to the optimization of impulsive trajectories, which can be described by a limited number of variables. In this case, the number of function evaluations that are required to obtain the optimal solution is usually acceptable, whereas the large number of variables required to describe low thrust trajectories with sufficient accuracy makes the use of EA for this kind of problems less attractive. Several studies concerning the optimization of impulsive interplanetary trajectories with gravity assists by means of EAs have been published in the recent literature; tuning of the algorithm parameters to improve convergence and comparisons of the performance of the algorithms have been presented (Biesbroek 2003, 2006; Myatt et al. 2004; Di Lizia and Radice 2004; Vasile and De Pascale 2006; Bessette and Spencer 2006; Izzo et al. 2007; Vinko et al. 2007; Olds et al. 2007; Rosa Sentinella and Casalino 2009). These studies usually agree in stating that EAs are suitable means for the optimization of this kind of missions, even though their conclusions sometimes differ on determining which particular method allows for the best performance. Results may be affected by the choice of the parameters that rule the behavior of the optimization algorithms, and comparison of the results is sometimes difficult due to insufficient details about how the trajectory is modeled.

The tuning of the parameters is useful as far as it has a general validity and the algorithm performs well when applied to different problems. However, there is no guarantee that a particular setting can effectively deal with a new problem. In the present article, a hybrid evolutionary algorithm, which employs synergistically basic EAs, is presented and applied to the design of impulsive space trajectories. Thanks to the effort of different EAs, hybrid algorithms have a larger chance of performing well on different problems, with generic settings of the parameters and without any specific tuning; only population size and number of function evaluations should be varied to take the dimension of the problem (i.e., number of optimization variables) into account. The hybrid optimization procedure presented here adopts the synchronous island model (Tomassini 1995; Whitley et al. 1999), and runs three different optimizers (based on GA, DE, and PSO, respectively) "in parallel"; each algorithm acts on a separate population, but the best individuals found by each algorithm migrate to the others at prescribed intervals. Note that the synergistic use of different EAs has already been proposed for space trajectory optimization, e.g., in Vinko et al. (2007), but with an "in series" arrangement, which is less helpful when new problems are dealt with and the correct tuning of the algorithm parameters is not known. In comparison to existing methods, a "mass mutation" (MM) operator is also introduced: A high percentage of individuals (typically over 90%) is eliminated when the mean distance between the individuals is smaller than a prescribed value and the optimal objective function is stuck on the same value for more than a prefixed number of iterations (Rosa Sentinella 2007). This sort of re-initialization helps to avoid convergence to suboptimal solutions.

EAs are stochastic methods and the result typically changes each time the optimization procedure is repeated. Performance of the algorithms is evaluated in terms of efficiency (probability of success, i.e., capability of correctly finding the global optimum), which is more significant than best and/or average result and standard deviation typically considered in the existing literature concerning trajectory optimization, and computational effort (average number of function evaluations required to attain the optimum).

Impulsive rendezvous missions are considered to compare the performance of the hybrid and basic algorithms. First, the problem is dealt with by using GA, DE, and PSO separately to find the best settings for each algorithm. Then, the hybrid algorithm, with either optimal or nonoptimal settings of the parameters is tested. Different classes of Earth–Mars round-trip trajectories are finally optimized by means of the hybrid algorithm. Computations are carried out sequentially, but the algorithm is easily parallelizable.

2 Evolutionary algorithms

An EA is a population-based meta heuristic optimization algorithm which borrows its mechanisms from biological evolution: reproduction, mutation, recombination, natural selection and survival of the fittest. Candidate solutions to the optimization problem play the role of individuals in a population, and a cost or merit function determines how the individual is suited for the environment. Evolution of the population takes place with the repeated application of prescribed operators.

The present article deals with impulsive space trajectories. Each trajectory is defined by the real values of a fixed number of variables, that completely determine the trajectory (e.g., relevant dates and corresponding positions, magnitude and orientation of each impulse). The variables can assume values in user-specified ranges. The mission ΔV is the function to be minimized.

An initial population of randomly generated candidate solutions comprise the first generation. The fitness function, in this case the inverse of the mission ΔV, is evaluated for the candidate solutions. Subsequent generations are created with a bias toward higher fitness. The new candidates compete with old individuals for their place in the next generation (survival of the fittest). This process is repeated until a solution with sufficient quality is found or a previously determined computational limit (number of generations or number of function evaluations) is reached. Similar techniques, which differ in the implementation details and the nature of the particular applied problem, have been proposed. An optimization code which employs GA, DE and PSO, described in detail in Rosa Sentinella (2008), has been developed and is used in this article.

2.1 Genetic algorithm

Genetic algorithms (Holland 1975; Mitchell 1996), the most popular type of EA, work on a population of potential solutions applying the principle of survival of the fittest to produce better and better approximations of the optimal solution. At each iteration, a new set of solutions is created by selecting parent individuals according to their fitness and breeding them together using operators borrowed from natural genetics. This process leads to the evolution of the population, with new individuals which are better suited to the environment than the individuals that they were created from, just as in natural adaptation (Goldberg 1997; Goldberg and Deb 1991).

A starting population of N_i individuals is created by means of a random number generator with uniform distribution. Each individual is characterized by the real values of N_p optimization variables. For each individual, the mission ΔV, which is the objective function here minimized, is evaluated; a large value is associated to the combinations that do not allow for a solution. The fitness function is defined as the inverse of ΔV. In the proposed algorithm, the *extended random initialization* (ERI) procedure of Bramlette (1991) is introduced, whereby random initializations are tried for each individual until a significant solution is found or a maximum number of tries is reached. This initial effort in terms of computational cost is repaid with a faster convergence to the global optimum, which is reached after a lower number of function evaluations (Rosa Sentinella 2007).

Genetic operators are then used iteratively to get an improved population. The principle of *Elitism* is first used to avoid the loss of good individuals caused by other genetic operators (mutation, cross-over). The best individuals, i.e., those with the highest fitness (lowest ΔV), are stored and replace the worst ones of the following generation; in the present work 3 individuals are selected and saved during each iteration.

The new generation is obtained by first forming a "parent" population, which emphasizes the good solutions and eliminates the bad solutions. The GA optimization tool that has been developed can use three different types of *Selection operator*:

1. Tournament selection: Two individuals are randomly chosen and compared to each other; the best is placed into the parent population. This procedure is repeated systematically, and each individual participates in two tournaments, so it could be present twice in the parent population if it wins both tournaments, or it could be eliminated if it loses twice. The tournament selection has faster convergence, and lower computational time and complexity compared to any other selection operator that exists in the literature (Goldberg and Deb 1991), but can become stuck very easily in local minima.
2. Roulette wheel selection: The probability of each individual to be selected for reproduction is chosen to be proportional to the fitness value. The basic roulette wheel selection method is stochastic sampling with replacement, and it can have potentially an unlimited spread, because any individual with a positive fitness function could entirely fill the parent population.
3. Stochastic universal sampling: This operator is an evolution of the previous one, with the aim to reduce the spread of the parent population. Instead of the single selection pointer employed in roulette wheel methods, the stochastic universal sampling uses N equally spaced pointers, where N is the number of selections required.

Cross-over operator is then applied to get a new generation of individuals: couples of individuals are randomly selected among the parent population so as each individual is chosen once. In the real-coded formulation, cross-over is applied to each variable according to a probability distribution defined as:

$$P(\beta) = \begin{cases} 0.5(\eta + 1)\beta^{\eta} & \text{for} \quad \beta \leq 1 \\ 0.5(\eta + 1)/\beta^{(\eta+2)} & \text{otherwise} \end{cases} \tag{1}$$

The procedure determines the new variable values y_1 and y_2 from those of the parent individuals x_1 and x_2; a random number $0 \leq u \leq 1$ is first chosen, $\bar{\beta}$ is determined so as $\int_0^{\bar{\beta}} P(\beta)\mathrm{d}\beta = u$ and the "children" are computed: $y_1 = 0.5[(x_1 + x_2) - \bar{\beta}|x_2 - x_1|]$ and $y_2 = 0.5[(x_1 + x_2) + \bar{\beta}|x_2 - x_1|]$. The parameter η controls how close children solutions are with respect to parent solutions (the larger is η, the closer are the solutions). Two settings have been tried: constant-η ($\eta = 2$ is assumed as suggested by Deb) or variable-η

($\eta = 2, 3, 4,$ and 5 are used during the first, second, third and fourth quarter of function evaluations, respectively).

Mutation is applied to the new population to increase the number of explored solutions and keep diversity in the population. Some of the variables are changed, according to a small specified probability (2% is used in the present article; results are only slightly affected by the chosen value); the new value is chosen randomly in the specified range of the variable. The objective function of each individual of the new generation is finally evaluated; the worst ones are discarded and replaced by the elite individuals of the former generation. The whole procedure is repeated for a fixed number of generations N_g or until a prefixed number of function evaluations is reached.

2.2 Differential evolution

DE (Storn and Price 1995; Storn 1996) is a parallel direct search method which utilizes a population of N_i individuals of N_p dimension for N_g generations, just as GA does. ERI can be optionally added as well, to initialize the population. DE generates new vectors of variables by adding the weighted difference between two population vectors to a third one. If the resulting individual exhibits a higher fitness than a predetermined population member, in the next generation the new individual replaces the one it was compared to; otherwise, the old individual is retained. This basic principle can be varied, and in fact there are several practical variants of DE; according to Storn (1996), the following strategies can be used in the developed algorithm:

1. DE/best/1:
   $y_{i,G+1} = x_{\text{best},G} + F(x_{r2,G} - x_{r3,G})$
2. DE/rand/1:
   $y_{i,G+1} = x_{r1,G} + F(x_{r2,G} - x_{r3,G})$
3. DE/rtb/2:
   $y_{i,G+1} = x_{r1,G} + F(x_{\text{best},G} + x_{r2,G} - x_{r3,G})$
4. DE/best/2:
   $y_{i,G+1} = x_{\text{best},G} + F(x_{r1,G} + x_{r2,G} - x_{r3,G} - x_{r4,G})$
5. DE/rand/2:
   $y_{i,G+1} = x_{r5,G} + F(x_{r1,G} + x_{r2,G} - x_{r3,G} - x_{r4,G})$
6. DE/rtb/1:
   $y_{i,G+1} = x_{r1,G} + C(x_{\text{best},G} - x_{r1,G}) + F(x_{r2,G} - x_{r3,G})$

where F and C are scaling factors, $x_{rj,G}$ $(j = 1, 4)$ are vectors chosen randomly, and $x_{\text{best},G}$ is the best vector at the G-th iteration. These six strategies are tested here, with $F = 0.5$, $C = 0.9$ and cross-over probability (Storn and Price 1995; Storn 1996) fixed at 0.9.

2.3 Particle swarm optimization

PSO is a population based stochastic optimization technique, inspired by social behavior of bird flocking or fish schooling (Kennedy and Eberhart 1995; Eberhart and Kennedy 1995). The system is initialized with a population of random solutions and searches for the optimum by updating generations. However, unlike GA or DE, PSO has no operators like crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles.

The original intent was to graphically simulate the choreography of a bird flock or fish school, where each individual follows the one which is nearest to the food. PSO learns from this scenario and uses it to solve optimization problems. In PSO, each single solution is a "bird" in the search space, and is called "particle". The values of the N_p optimization variables define the particle position; a velocity vector is also associated to each particle. The particles fly through the problem space by following the particle with the best fitness function. PSO is initialized with a group of random particles; each particle moves according to its instantaneous velocity, which is updated at each iteration in order to improve the solution. Two criteria are used: The velocity is changed to move the particle toward the personal best P_{best}, i.e., the best solution that the particle has reached in the previous iterations (cognitive acceleration), and toward the global best G_{best}, the best solution that any particle in the population has reached (social acceleration). The particle updates its velocity and positions according to the relations

$$v = v + c_1 \cdot k_1 \cdot (P_{\text{best}} - p) + c_2 \cdot k_2 \cdot (G_{\text{best}} - p) \tag{2}$$

$$p = p + v \tag{3}$$

where v is the particle velocity, p is the particle position, k_1 and k_2 are random numbers in the [0,1] interval. The learning factors c_1 and c_2 depend on the adopted strategy: Common, Trelea type 1 and 2 and Clerk's strategies are tested in the present article (Trelea 2003). Particles' velocities are limited to a maximum value V_{max} equal to one quarter of the variable range.

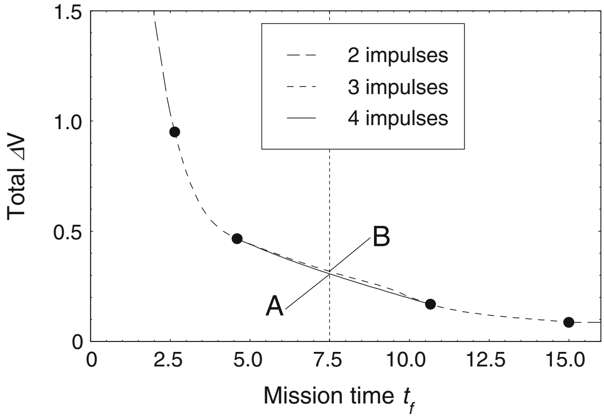
Compared with GAs and DEs, the information sharing mechanism in PSO is significantly different. In GAs and DE, individuals share information with each other, so the whole population moves like a single group toward an optimal area. In PSO, only one solution G_{best} shares information with others, with a one-way information sharing mechanism. Compared with GA, all the particles tend to converge to the best solution more quickly, in most cases.

3 Multiple-impulse rendezvous mission

The problem of space trajectory optimization considered here is an impulsive time-fixed rendezvous transfer between two bodies with circular coplanar orbits (Prussing and Chiu 1986; Colasurdo and Pastrone 1994). Variables are made nondimensional by using the radius of the initial orbit and the corresponding circular velocity as reference values. The chaser is on an orbit with $r_i = 1$ and the target is on an orbit with $r_f = 1.2$; the initial phase angle between them is $\Delta\theta = \pi$ (i.e., 180°).

The optimum ΔV is presented in Fig. 1 as the function of the allowable time t_f. If the time is scarce, a two-impulse maneuver results to be optimal, but at $t_f \approx 2.6$ a three-impulse and then at $t_f \approx 4.6$ a four-impulse maneuver become optimal. If the time constraint becomes less stringent and enough time allows for waiting for the correct spacecraft phasing, the number of impulses diminishes and initial and final coast arcs appear in order to eventually reach the unconstrained global optimum, i.e., the Hohmann transfer. At $t_f \approx 10.7$ the three-burn transfer becomes again optimal, with an initial coast arc at $t_f \approx 13.7$. At $t_f \approx 15.0$ a two-impulse solution with a long initial coast arc takes over; at $t_f = 15.138$ the Hohmann transfer becomes available; the chaser remains on the initial orbit for $\Delta t = 11.514$ (until the optimal phasing, with the target 22.1° ahead of the chaser, is attained) and then performs the two-impulse Hohmann transfer which lasts $\Delta t = 3.624$.

Fig. 1 ΔV for multi-impulse rendezvous maneuver, $r_f = 1.2$, $\Delta\theta = \pi$



Six variables define the trajectory: Magnitude and direction of the impulse at $t = 0$ (i.e., first impulse), true anomaly between first and second impulse, magnitude and direction of the second impulse, true anomaly between second and third impulse. The final two impulses are computed by solving Lambert's problem for the transfer that joins the position at the third impulse and the target position at t_f in the remaining time. The impulse magnitudes vary between 0 and $\sqrt{2} - 1$ (i.e., the value required to escape from the initial orbit); the impulse direction is free, and the impulse angle above horizon varies between $-\pi$ and π; the true anomaly change between impulses can vary between 0 and 4π (up to two revolutions are allowed). The initial population is chosen randomly, and many individuals do not allow for a solution of Lambert's problem, e.g., if they provide a negative remaining time for the final leg; this problem is particularly true for the shortest time of flight. Bramlette's ERI has been introduced to increase the number of good solutions in the initial population.

The mission with $t_f = 7.5$ is chosen to evaluate the performance of the optimization procedure. For this time-length, two close local optima exist (points A and B in Fig. 1). The global minimum corresponds to a four-impulse mission with $\Delta V = 0.3065$, but a locally-optimal three-impulse mission ($\Delta V = 0.3175$) also exists; these trajectories are shown in Fig. 2. Several other locally optimal solutions with worse performance can also be found, either with three or four impulses. Each simple optimizer has been systematically run $n = 40$ times to test the behavior when dealing with this problem and the capability to correctly locate the global optimum. The unknown probability of success of each algorithm ϵ is estimated as k/n, where k is the number of successful tries (best result within 1% of the global optimum). EAs are stochastic methods and a binomial distribution $p_k = C(n,k)\epsilon^n(1-\epsilon)^n$, where $C(n,k)$ indicates the binomial coefficient, can be assumed for k. The binomial distribution is approximated by the normal distribution and, under this assumption, the error made by assuming $\epsilon = k/n$ is $e \le \sqrt{\epsilon(1-\epsilon)Z_{1-\alpha/2}/n} < 0.5\sqrt{Z_{1-\alpha/2}/n}$ (with confidence $1-\alpha$), where Z is the normal distribution percentile. For a 95% confidence, $Z_{0.975} = 1.96$ and the error is about ±10%, which is considered to be sufficient for a preliminary evaluation of the algorithms.

With a maximum number of 600000 function evaluations, 3 different combinations of number of individuals and number of iterations ($N_i = 60$ and $N_g = 10000$, $N_i = 100$ and $N_g = 6000$, $N_i = 200$ and $N_g = 3000$) were tested for GA and DE, and 2 combinations for PSO ($N_i = 24$ and $N_g = 25000$, $N_i = 30$ and $N_g = 20000$). For these combinations, each optimizer is tested in his simplest form (S), with ERI (E, 200 initial tentatives for each

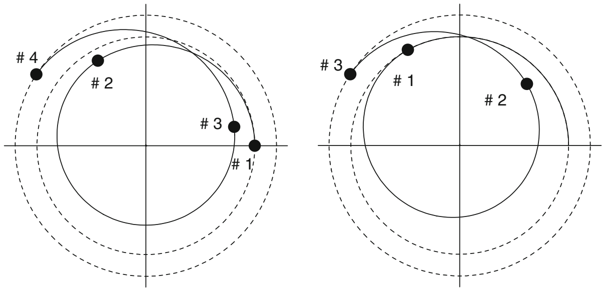


Fig. 2 Four-impulse (global optimum) and three-impulse (local optimum) missions for $t_f = 7.5, r_f = 1.2, \Delta\theta = \pi$

individual), and with both ERI and MM (EM), with a “kick” every 50 stuck iterations. All the strategies for each optimizer have been tested, to identify the best strategy and parameter setting for each basic algorithm. Then, the hybrid cooperative algorithm is used with either the optimal settings or with a nonoptimal choice of parameters for the basic algorithms. Information is shared between the algorithms by letting the best 6 individuals to be cloned and migrate to the other subpopulations every 20 iterations; the adopted values are purposely not optimized, to demonstrate the method applicability to new problems without any specific tuning. Additional tests are finally executed with different time of flight t_f.

3.1 Results of basic algorithms

The results of the multiple optimization runs show that generally DE works better than GA and PSO for this kind of problem, in agreement with existing literature. The results for DE with a population of 60 individuals are presented in Table 1, which shows the minimum attained velocity change $\Delta V_{\min}$, its average value ΔV_{mean} and standard deviation S, and the algorithm efficiency, i.e., probability of success. The introduction of ERI and MM typically provides a remarkable benefit. The best performance in terms of efficiency (up to 100% with ERI and MM) is obtained by adopting strategies DE/rand/1 and DE/rand/2, whereas the worst performance is for the DE/best/1 strategy (as low as 20% with 60 individuals without ERI and MM). The algorithm efficiency usually shows a slight improvement when larger population sizes are tested, but the best strategies reach 100% efficiency even with the smallest population when ERI and MM are introduced; on the other hand, the number of function evaluations which must be performed, on average, to attain the minimum grows remarkably when N is increased (for instance, when DE/rand/1 is employed with ERI and MM, it grows from 79000 with 60 individuals to 112000 with 200 individuals), and a 60-individual population is deemed suitable for this problem.

The results for the GA algorithm with a population of 200 individuals are shown in Table 2. In this case, the largest population is to be preferred; only the algorithm employing tournament selection finds the solution (with a very small 2.5% efficiency) when a population of 60 individuals is adopted; with 100 individuals, a good efficiency (up to 92.5) is obtained when tournament selection is used, whereas roulette wheel never finds the solution and stochastic universal sampling has a maximum efficiency of 7.5%. The GA seems to point to the correct optimum (the average value is close to the global optimum), but the algorithm convergence is too slow, and the minimum cannot be reached before the prescribed 600000 function evaluations, when a small population is adopted. It appears that the increase in the diversity of the initial population is beneficial to the convergence process and 200 individuals are used in the following. The results confirm that tournament selection is the most efficient selection

Table 1 Results for DE 60 individuals $F = 0.5$ CR $= 0.9$ and $t_f = 7.5$, $r_f = 1.2$, $\Delta\theta = \pi$

Strategy	ΔV_{min}	ΔV_{mean}	S	$\epsilon(\%)$
DE/best/1 S	0.3065	0.3284	4×10^{-3}	20
DE/best/1 E	0.3065	0.3281	2×10^{-2}	35
DE/best/1 EM	0.3065	0.3105	6×10^{-4}	85
DE/rand/1 S	0.3065	0.3227	4×10^{-3}	47.5
DE/rand/1 E	0.3065	0.3073	1×10^{-4}	95
DE/rand/1 EM	0.3065	0.3065	1×10^{-6}	100
DE/rtb/2 S	0.3065	0.3127	1×10^{-3}	65
DE/rtb/2 E	0.3065	0.3140	1×10^{-3}	67.5
DE/rtb/2 EM	0.3065	0.3066	1×10^{-6}	100
DE/best/2 S	0.3065	0.3162	1×10^{-3}	55
DE/best/2 E	0.3065	0.3169	2×10^{-3}	57.5
DE/best/2 EM	0.3065	0.3093	4×10^{-4}	80
DE/rand/2 S	0.3065	0.3069	5×10^{-5}	97.5
DE/rand/2 E	0.3065	0.3065	7×10^{-7}	100
DE/rand/2 EM	0.3065	0.3066	2×10^{-6}	100
DE/rtb/1 S	0.3065	0.3124	9×10^{-4}	80
DE/rtb/1 E	0.3065	0.3095	5×10^{-4}	82.5
DE/rtb/1 EM	0.3065	0.3066	7×10^{-6}	100

operator, and the benefit provided in most cases (at least, those with the best performance) by the introduction of ERI and, in particular, MM. The number of function evaluations required, on average, to reach the minimum is typically around 100000–150000, comparable to that of the DE algorithm. Variable η (v-η) shows slightly better performance than constant η (c-η).

The results of the PSO algorithm are shown in Table 3 for a population of 30 individuals (a slight decrease in terms of efficiency is found with a 24-individual population). Common (Co) and Trelea type 2 (T2) strategies exhibit better performance than Trelea 1 (T1) and Clerc's (Cl) strategies. Poor efficiency affects the performance of the algorithm, but a very small number of function evaluations is required to reach the minimum in the successful runs (typically 50000, on average, but as low as 10000 in certain cases), when the algorithm finds the right ”direction”. This peculiarity makes PSO very useful for hybrid algorithms, as PSO can reach the optimum after few iterations when it receives a good starting solution from the other algorithms. Again, ERI and more remarkably MM provide a significant benefit, which is even larger compared to DE and GA, as PSO seems to be unable to allow for convergence on any minima without initial good individuals.

In summary, the results show that, for a strategy characterized by slow convergence, which needs a large number of iterations to “mold” the population, the MM operator may be less beneficial, as it may disrupt the population, whereas MM always significantly improves the efficiency of a high-performance algorithm. One could search for the right compromise between maximum number of stuck iterations allowed and number of individuals to be “killed”, but this is not an easy task and may be strongly dependent on the particular problem that is dealt with and on the adopted algorithm. GA needs larger populations, whereas DE and PSO work well even with a small number of individuals. An excessive increase

Table 2 Results for GA 200 individuals and $t_f = 7.5$, $r_f = 1.2$, $\Delta\theta = \pi$

Strategy	ΔV_{min}	ΔV_{mean}	S	$\epsilon(\%)$
GA/T/c-η S	0.3065	0.3085	3×10^{-4}	82.5
GA/T/c-η E	0.3065	0.3094	2×10^{-3}	77.5
GA/T/c-η EM	0.3065	0.3068	4×10^{-5}	97.5
GA/T/v-η S	0.3065	0.3092	4×10^{-4}	85
GA/T/v-η E	0.3065	0.3086	3×10^{-4}	82.5
GA/T/v-η EM	0.3065	0.3066	3×10^{-6}	100
GA/R/c-η S	0.3065	0.3216	1×10^{-3}	12.5
GA/R/c-η E	0.3065	0.3246	3×10^{-3}	7.5
GA/R/c-η EM	0.3065	0.3131	9×10^{-4}	5
GA/R/v-η S	0.3065	0.3129	1×10^{-3}	5
GA/R/v-η E	0.3065	0.3225	2×10^{-3}	5
GA/R/v-η EM	0.3065	0.3244	2×10^{-3}	10
GA/S/c-η S	0.3065	0.3153	1×10^{-3}	25
GA/S/c-η E	0.3065	0.3130	2×10^{-3}	15
GA/S/c-η EM	0.3065	0.3099	4×10^{-4}	57.5
GA/S/v-η S	0.3129	0.5705	3×10^{-2}	0
GA/S/v-η E	0.3065	0.3147	5×10^{-4}	25
GA/R/v-η EM	0.3066	0.3066	2×10^{-3}	100

Table 3 Results for PSO 30 individuals and $t_f = 7.5$, $r_f = 1.2$, $\Delta\theta = \pi$

Strategy	ΔV_{min}	ΔV_{mean}	S	$\epsilon(\%)$
PSO/Co S	0.3474	0.8536	4×10^{-2}	0
PSO/Co E	0.3066	0.4161	2×10^{-2}	2.5
PSO/Co EM	0.3065	0.3286	2×10^{-3}	27.5
PSO/T1 S	0.3477	0.5465	2×10^{-1}	0
PSO/T1 E	0.3065	0.3737	5×10^{-3}	7.5
PSO/T1 EM	0.3065	0.3331	2×10^{-3}	10
PSO/T2 S	0.3477	0.4152	3×10^{-2}	0
PSO/T2 E	0.3175	0.3953	8×10^{-3}	0
PSO/T2 EM	0.3065	0.3291	4×10^{-3}	30
PSO/Cl S	0.3477	0.7180	5×10^{-3}	0
PSO/Cl E	0.3084	0.4252	1×10^{-2}	0
PSO/Cl EM	0.3093	0.3876	7×10^{-3}	0

in the population size is paid in terms of number of function evaluations, without significant improvements in terms of efficiency. In the end, for this problem, the best strategies are DE/rand/2 with 60 individuals for DE, tournament selection, crossover with variable η with 200 individuals for GA, and Trelea's type 2 with 30 individuals for PSO, all with ERI and MM.

3.2 Results of hybrid algorithms

The developed multi-population hybrid algorithm is used to exploit the peculiarities of GA, DE and PSO, with the aim of reducing the number of function evaluations and increasing the efficiency of the procedure. First, all the hybrid algorithms, which can be built by differently combining the basic algorithms, are tested for $t_f = 7.5$ with the optimal set of parameters previously found. Then, the complete hybrid algorithm with DE, GA, and PSO is tested on critical points of the curve in Figure 6.1. The most significant results are shown in Tables 4 and 5, respectively, which show the efficiency ϵ, the minimum N_{min} and average N_{mean} number of functions evaluations to attain the minimum, and the corresponding standard deviation S_N. The minimum ΔV is also shown in Table 5. Finally, to show the capabilities of the hybrid algorithm, the same tests are carried out by adopting non-optimal settings for the parameters of the basic algorithms; in particular, DE/best/2, with 60 individuals, GA with stochastic universal sampling, crossover with constant η and 200 individuals, and Trelea's type 2 with 30 individuals for PSO, with ERI and MM are adopted (one should note that, even though this PSO strategy is actually the best available choice, it has only a 30% efficiency). Results are shown in Tables 6 and 7, respectively.

At $t_f = 7.5$ all the hybrid algorithms find the solution with a 100% efficiency, as one can expect since also the basic algorithms, at least DE and GA, always locate the global optimum. Table 4 compares the performance in terms of function evaluations, also showing the results for the basic DE, GA, and PSO algorithms. The computational cost, i.e., number of function evaluations to reach the minimum, is comparable with that of the simple algorithms. Only the combination of DE anf PSO has a lower computational cost, thanks to the DE capability of

Table 4 Performance of the hybrid algorithms with optimal settings $t_f = 7.5, r_f = 1.2, \Delta\theta = \pi$

Algorithm	ϵ(%)	N_{min}	N_{mean}	S_N
DE	100	37500	183463	13254
GA	100	73198	139138	10577
PSO	30	9022	196702	25690
GA + DE	100	82338	122654	19067
GA + PSO	100	72793	154396	12423
DE + PSO	100	28069	80188	14623
GA + DE + PSO	100	72097	135516	13368

Table 5 Performance of the complete hybrid algorithm with optimal settings, $r_f = 1.2, \Delta\theta = \pi$

t_f	ΔV_{min}	ϵ (%)	N_{min}	N_{mean}	S_N
2.4	1.1039	100	58291	60978	2079
5	0.4406	100	98989	195135	17534
7.5	0.3065	100	72097	135516	13368
10.7	0.1666	100	67900	157275	13535
15	0.0878	100	51489	56149	3063
15.138	0.0869	100	50159	56858	3446

Table 6 Performance of the hybrid algorithms with nonoptimal settings—$t_f = 7.5$, $r_f = 1.2$, $\Delta\theta = \pi$

Algorithm	ϵ (%)	N_{min}	N_{mean}	S_N
DE	80	20884	67122	36268
GA	57.5	71021	227527	1870
PSO	30	9022	196702	25690
GA + DE	97.5	101426	146743	24578
GA + PSO	87.5	62429	427379	14353
DE + PSO	90	26938	57064	24922
GA + DE + PSO	100	97048	177441	18595

Table 7 Performance of the complete hybrid algorithm with nonoptimal settings—$r_f = 1.2$, $\Delta\theta = \pi$

t_f	ΔV_{min}	ϵ (%)	N_{min}	N_{mean}	S_N
2.4	1.1039	100	58246	59786	920
5	0.4418	100	115270	244433	10470
7.5	0.3065	100	97048	177441	18595
10.7	0.1666	97.5	66501	119858	14913
15	0.0878	100	50591	54702	2294
15.138	0.0869	100	50167	54631	2496

locating the region that contains the global optimum and the fast convergence of PSO when the rigth direction is found.

Table 5 shows that, when the hybrid algorithm is used with an optimal parameter setting, the optimum is again found in 100% of the tries for all the considered cases. The computational cost is larger for those points where local optima with similar ΔV exist, i.e., for $5 < t_f < 11$, whereas the required number of function evaluations is quite small in the other cases.

One cannot know “a priori” the strategies and settings that should be adopted when dealing with a new problem, and must either count on experience or carry out an accurate and time-consuming analysis of the optimizers’ behavior on that particular problem, if enough time is available. If this is not possible, the optimizers must be used with a ”standard” strategy and set of parameters, which could be not suitable for the particular problem that is considered. Results in Table 6 show that, even using the hybrid algorithm with a nonoptimal set of parameters, one can obtain an efficiency that is as high as the one obtained with the optimal settings, and that the efficiency of the hybrid algorithm is higher than that of the simple algorithms. In particular, it is remarkable that the synergistic use of GA and PSO, which present a 57.5 and 30% efficiency, respectively, when used separately, produces an efficiency of almost 90%, thanks to the exchange of information between the algorithms; the good capability of the GA to locate the region that contains the optimum and the fast convergence of the PSO algorithm almost double the convergence efficiency. It is also worthwhile to note that again the DE+PSO combination presents the fastest convergence, but only the complete algorithm is able to obtain a 100% efficiency. The complete hybrid algorithm with this nonoptimal setting performs very well in all the critical points, always obtaining the correct solution but in one case for $t_f = 10.7$.

The point at $t_f = 13.8$ is peculiar as two three-impulse missions with close performance exist. The global optimum corresponds to a mission with an initial coast arc and has $\Delta V = 0.102024$, whereas a different mission, which starts with an impulse, has $\Delta V = 0.102040$. The hybrid algorithm with optimal settings was tested performing $n = 200$ runs to improve the accuracy of the efficiency estimation to 5%. The algorithm always finds either solution (the average number of function evaluations is 105615, with a minimum of 8147), but the global optimum is found in only 55% of the tries. The ΔV of the solutions seem too close to each other and it is extremely difficult to locate the correct solution. Rather curiously, better performance are obtained with the hybrid algorithm with nonoptimal settings; again, the algorithm always finds either solution, but the global optimum is obtained in 67.5% of the tries, with a slightly larger number of function evaluations (average 150211, minimum 7274).

4 Earth–Mars round trip missions

Evolutionary algorithms are particularly suited to deal with interplanetary trajectories with impulsive maneuvers and planetary flybys, as they can explore large search space in a relatively low time. However, due to the heuristic nature of the algorithms and the complexity of the problems, they may present a low efficiency in finding the correct solution, and multiple runs are typically required (Olds et al. 2007; Rosa Sentinella and Casalino 2009). Earth–Mars round-trip trajectories are here analyzed to show the capabilities of the hybrid algorithm. No specific tuning of the parameters is implemented; 200, 60, and 30 individuals are used for GA, DE, and PSO, respectively, with the optimal settings found for the rendezvous problem, and 20 tries per individual are carried out during the ERI. The spacecraft leaves a 300-km circular low Earth orbit and arrives on a 1-sol elliptical Mars orbit with a 3800 km periapsis. The same parking orbits are assumed for the return leg; favorable orientation of the parking orbits is always supposed. The patched-conic approximation is adopted and the ΔVs at departure and arrival of each leg are $\Delta V = \sqrt{V_\infty^2 - V_{\mathrm{esc}}^2} - V_{\mathrm{orb}}$, where V_∞^2 is the hyperbolic excess velocity, V_{esc} is the escape velocity and V_{orb} is the orbital velocity at the impulse, which occurs at the orbit periapsis.

Three classes of missions are considered (Walberg 1993; Casalino et al. 1998). Conjunction-class missions are minimum-ΔV missions characterized by long trip time (about 1000 days); they comprise two Hohmann-like transfers joined by a long stay on Mars to wait for suitable phasing between the planets to start the return leg. Mission opportunities occur every Earth–Mars synodic period (2.13 years), when the planets present the same relative positions; the mission cost and time-length change due to Mars' eccentricity and inclination and Earth's eccentricity. After a syzygistic period, which encompasses 15 synodic periods (32 years), the planets have the same absolute positions and a mission can be repeated almost exactly. Reference is made to the results of Casalino et al. (1998) and a 32-year interval starting on January 1, 2000 is here considered.

Ballistic legs are first considered. The mission has four optimization variables, i.e., Earth departure date, time-length of outbound and inbound leg (varying between 1 and 300 days), and stay time (between 1 and 500 days). Lambert's problem is solved for each leg, and the required hyperbolic excess velocities at departure and arrival are easily computed. The hybrid algorithm always finds the correct optimal solution when the Earth departure date is let vary in a single synodic period. The capability of locating the globally optimal mission among the fifteen local optima (with ΔV ranging from 8.974 to 9.699 km/s) when the Earth departure date is let vary in the selected 32-year interval is investigated by running the

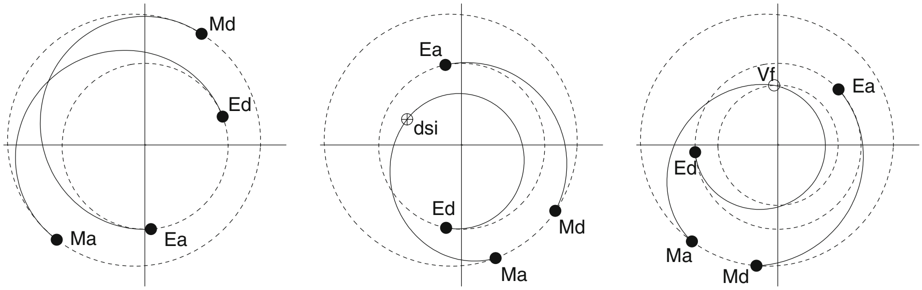


Fig. 3 Best opportunities for conjunction-class, opposition-class (outbound impulse) and Venus-flyby (outbound) in the 2000–2031 time frame. Earth departure (Ed), Mars arrival (Ma), Mars departure (Md), Earth arrival (Ea), Venus flyby (Vf), and deep-space impulse (dsi) are highlighted

algorithm 200 times. Convergence to the global optimum (shown in Fig. 3, departure in 2009, $\Delta V = 8.974$ km/s, Mars arrival and departure are close to the line of nodes) is obtained in 75% of the tries, whereas the algorithm provides the second-best solution (departure in 2025, $\Delta V = 8.988$ km/s) in 24% of the tries and the fourth-best solution (departure in 2007, $\Delta V = 9.122$ km/s) in 1% of the tries. The possibility of adding a deep-space impulse during each leg is also considered for the sake of completeness. For each impulse, four optimization variables are added, namely, the time from leg departure to the impulse devided by the leg time-length, and the hyperbolic excess velocity components at departure in a radial/tangential/normal reference frame (they are let vary between ± 6 km/s). No improved solutions were found, signaling that the deep-space impulses are actually not required. However, the larger number of variables (12) reduces the algorithm performance and the best solution is found in 37% of the tries (the second-best in 17%, the remaining 46% shared between 8 other solutions).

For the first human missions to Mars, the long stay time of conjunction-class missions will not be acceptable. If the stay time is shortened, either leg must be modified to make up for the nonfavorable phasing; the spacecraft must fly inside the Earth's orbit, where an impulse must be used close to the perihelion to correct the spacecraft velocity in order to encounter the target planet. The impulse can occur either during the Earth–Mars leg, i.e., the outbound leg, or during the return trip, i.e., the inbound leg. Seven parameters (Earth departure date, leg lengths and four variables for the deep-space impulse) define the opposition-class mission, when only one impulse is allowed. In a 32-year syzygistic cycle, 15 opportunities with impulse in the outbound leg and 15 with the impulse in the inbound leg can be found. In the first case, the length of the outbound leg is allowed to vary between 1 and 400 days, wheres the inbound leg lasts between 1 and 300 days; the opposite is assumed when the impulse occurs in the inbound leg. A maximum trip time of 700 days is fixed and a 60-day stay time on Mars is here considered. The ΔV ranges from 15.392 to 20.012 km/s. The optimization algorithm correctly identifies the global optimum for missions with the outbound deep-space impulse (shown in Fig. 3, departure in 2019, $\Delta V = 15.516$ km/s, Mars intercepted close to the perihelion) in 53% of the tries and converges to the second best solution in 21% of the tries (departure in 2004, $\Delta V = 15.631$ km/s), with the rest shared between other locally optimal solutions (ΔV always lower than 16.2 km/s). Among these solutions, a longer mission with the same 2019 departure as the global optimum and $\Delta V = 16.2$ km/s is found in 5% of the tries; however, the global optimum is attained in 100% of the tries if the departure date is constrained in 2019. The algorithm performance on the optimization of missions with the inbound impulse are slightly worse, as the global optimum (departure in

2018, $\Delta V = 15.380$ km/s) is only found in 24% of the tries, with 30% of the tries converging to the solution with departure in 2016 ($\Delta V = 15.679$ km/s) and 15% to the solution with departure in 2001 (15.392 km/s).

The midcourse impulse usually occurs close to the orbit of Venus, and Venus flyby can efficiently replace the impulse when the planet is in a favorable position. A syzygistic cycle contains seven Venus–Mars synodic periods and a favorable position between these planets occurs seven times. However, the Earth occupies different positions at each opportunity, and only three chances can actually be exploited. In one case, Venus position is extremely favorable, and a mission with very low ΔV can be found. Two less-performing missions can be found in each of the remaining occasions, producing five missions with the flyby during the outbound leg; additional five symmetrical missions can be found with the flyby during the inbound leg. The ΔV ranges from 12.124 to 23.283 km/s. An additional impulse is allowed during the leg without flyby, and the number of optimization variables is 8 (departure date, time-lengths of the three heliocentric legs, and four variables to define the impulse). Each leg can vary between 1 and 250 days. For the outbound flyby mission, the global optimum (shown in Fig. 3, departure in 2017, $\Delta V = 12.124$ km/s) is correctly located in 75% of the tries; the algorithm has more difficulties in finding the global optimum (departure in 2001, $\Delta V = 12.173$ km/s) when the flyby occurs during the inbound leg, probably because the optimum is at the beginning of the considered time-frame, and solutions located in the middle of the search space seem to have a larger chance of attracting the solution (e.g., the second best solution with departure in 2013 and $\Delta V = 12.624$ km/s is found in 41% of the tries versus 33% of the global optimum). Missions with inbound flyby and departure in 2002 are particularly interesting as two close optima exist: the mission with an additional impulse during the outbound leg has a slightly better performance compared to the mission without deep-space impulse [14.744 vs. 14.758 km/s, note that a printing error affects the referenced article by Casalino et al. (1998)]. A 68% efficiency in finding the correct optimum with the additional impulse has been found when the departure date was constrained in 2002.

5 Conclusion

An hybrid evolutionary algorithm, which uses differential evolution, genetic algorithm and particle swarm optimization in a cooperative way to optimize impulsive spacecraft trajectories has been developed and applied to a rendezvous problem and interplanetary missions. The method shares information between the basic algorithms, which work “in parallel” according to the island model, by letting the best individuals migrate to the other algorithms at prescribed intervals. The hybrid algorithm shows better performance in terms of efficiency compared to the basic algorithms also when nonoptimal settings are adopted, and therefore constitutes a viable means to deal with new problems, when the optimal parameter settings are not known. The results also show the benefit of introducing Bramlette’s random initialization procedure, to produce good individuals in the starting population, and, more remarkably, mass mutation, to allow for escaping from locally optimal solutions.

The use of the hybrid algorithm to deal with interplanetary trajectories with gravity assist and deep-space impulses, characterized by the existence of close minima, has shown that the algorithm usually presents a quite large efficiency in locating the correct global minimum, also when a large search space is explored. Few runs (and few minutes when a standard PC is employed) usually provide the certainty of obtaining the correct solution, thus proving that the developed algorithm is a useful tool for the preliminary analysis of complex interplanetary trajectories and for mission planning.

References

Bessette, C., Spencer, D.: Optimal space trajectory design: a heuristic-based approach. Adv. Astronaut. Sci. **124**, 1611–1628 (2006)

Biesbroek, R.: Study of genetic algorithm settings for trajectory optimisation. In: Paper Presented at the 54th International Astronautical Congress, Bremen, Germany, IAF-03-A.P.30, Sept.–Oct. (2003)

Biesbroek, R.: A comparison of differential evolution method with genetic algorithms for orbit optimisation. In: Paper Presented at the 57th International Astronautical Congress, Valencia, Spain, IAF-06-C1.4.02, Oct. (2006)

Bramlette, M.: Initialization, mutation, and selection methods in genetic algorithms for function optimization. In: Belew, R.K., Booker, L.B. (eds.) Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 100–107. Morgan Kaufmann, San Mateo, CA (1991)

Casalino, L., Colasurdo, G., Pastrone, D.: Mission opportunities for human exploration of Mars. Planet. Space. Sci. **46**(11/12), 1613–1622 (1998)

Colasurdo, G., Pastrone, D.: Indirect optimization method for impulsive transfer. In: Paper Presented at the AIAA/AAS Astrodynamics Conference, Scottsdale, AZ, AIAA 94-3762 (1994)

Conway, B.A., Chilan, C.M., Wall, B.J.: Evolutionary principles applied to mission planning problems. Celest. Mech. Dyn. Astron. **97**(2), 73–86 (2007)

Crain, T., Bishop, R., Fowler, W., Rock, K.: Interplanetary flyby mission optimization using a hybrid global-local search method. J. Spacecr. Rockets. **37**(4), 468–474 (2000)

Dachwald, B., Wie, B.: Solar sail kinetic energy impactor trajectory optimization for an asteroid-deflection mission. J. Spacecr. Rockets. **44**(4), 755–764 (2007)

Di Lizia, P., Radice, G.: Advanced global optimization tools for mission analysis and design. Final report of ESA Ariadna ITT AO4532/ 18139/04/NL/MV, Call 03/4101, ESA (2004)

Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Sixth International Synposium on Micro Machine and Human Science, pp. 39–43. IEEE, Piscataway, NJ, (1995)

Gage, P., Braun, R., Kroo, I.: Interplanetary trajectory optimization using a genetic algorithm. J. Astron. Sci. **43**(1), 59–76 (1995)

Goldberg, D., Deb, K. : A comparison of selection schemes used in genetic algorithms. In: Rawlins, G. (ed.) Foundations of Genetic Algorithms, vol. 1, pp. 450–457. Morgan Kaufmann, San Francisco, CA (1991)

Goldberg, D.: Genetic Algorithms in Engineering Design. Wiley, New York, NY (1997)

Hartman, J., Coverstone-Carroll, V., Williams, S.: Optimal interplanetary spacecraft trajectories via pareto genetic algorithm. J. Astron. Sci. **46**(3), 267–282 (1998)

Holland, J.: Adaption in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI (1975)

Izzo, D., Becerra, V., Myatt, D., Nasuto, S., Bishop, J.: Search space pruning and global optimization of multiple gravity assist spacecraft trajectories. J. Glob. Optim. **38**(2), 283–296 (2007)

Kennedy, J., Eberhart, R., Particle swarm optimisation. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 1942–1948. IEEE, Piscataway, NJ, (1995)

Mitchell, M.: Introduction to Genetic Algorithms. MIT Press, Ann Arbor, MI (1996)

Myatt, D., Becerra, V., Nasuto, S., Bishop, J.: Advanced global optimization tools for mission analysis and design. Final Report of ESA Ariadna ITT AO4532/18138/04/NL/MV, Call 03/4101, ESA (2004)

Olds, A., Kluever, C., Cupples, M.: Interplanetary mission design using differential evolution. J. Spacecr. Rockets. **44**(5), 1060–1070 (2007)

Prussing, J.E., Chiu, J.-H.: Optimal multiple-impulse time-fixed rendezvous between circular orbits. J. Guid. Control Dyn. **9**(1), 17–22 (1986)

Rauwolf, G., Coverstone-Carroll, V.: Near-optimal low-thrust orbit transfers generated by a genetic algorithm. J. Spacecr. Rockets. **33**(6), 859–862 (1996)

Rauwolf, G., Coverstone-Carroll, V.: Near-optimal low-thrust trajectories via micro-genetic algorithms. J. Guid. Control Dyn. **20**(1), 196–198 (1997)

Rosa Sentinella, M.: Comparison and integrated use of differential evolution and genetic algorithms for space trajectory optimisation. In: Proceedings of the 2007 IEEE Congress on Evolutionary Computation, pp. 973–978. IEEE Press, Singapore, (2007)

Rosa Sentinella, M.: Development of new procedures and hybrid algorithms for space trajectories optimisation. Ph.D. thesis, Politecnico di Torino, Turin, Italy (2008)

Rosa Sentinella, M., Casalino, L.: Genetic algorithm and indirect method coupling for low-thrust trajectory optimization. In: Paper Presented at the 42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Sacramento, CA, Paper AIAA 06-4468, June (2006)

Rosa Sentinella, M., Casalino, L.: Hybrid evolutionary algorithm for the optimization of interplanetary trajectories. J. Spacecr. Rockets. **46**(2), 365–372 (2009)

Storn, R.: On the Usage of differential evolution for function optimization. In: 1996 Biennial Conference of the North American Fuzzy Information Processing Society, pp. 519–523. NAFIPS, Berkeley, (1996)

Storn, R., Price, K.: Differential evolution—a simple and efficient adaptive scheme for global optization over continuos spaces. ICSI TR-95-012, ICSI (1995)

Tomassini, M.: A survey of genetic algorithm. In: Stauffer, D. (ed.) Annual Reviews of Computational Physics, vol. III, pp. 87–118. World Scientific, Singapore (1995)

Trelea, I.C.: The particle swarm optimization algorithm: convergence analysis and parameter selection. Inf. Process. Lett. **85**(6), 317–325 (2003)

Vasile, M., De Pascale, P.: Preliminary design of multiple gravity-assist trajectories. J. Spacecr. Rockets. **43**(4), 794–805 (2006)

Vinko, T., Izzo, D., Bombardelli, C.: Benchmarking different global optimisation techniques for preliminary space trajectory design. In: Paper Presented at the 58th International Astronautical Congress, Hyderabad, India, IAC-07-A1.3.01, Oct. (2007)

Walberg, G.: How shall we go to Mars? A review of mission scenarios. J. Spacecr. Rockets. **30**(2), 129–139 (1993)

Whitley, D., Rana, S., Heckendorn, R.B.: Exploiting separability in search: the island model genetic algorithm. J. Comput. Inf. Technol. **7**(1), 33–47, (1999) (special issue on evolutionary computing)

Woo, B., Coverstone-Carroll, V., Cupples, M.: Low-thrust trajectory optimization procedure for gravity-assist, outer-planet missions. J. Spacecr. Rockets. **43**(1), 121–129 (2006)