Journal of Guidance, Control, and Dynamics
Vol. 29, No. 6, November–December 2006

Engineering Notes

ENGINEERING NOTES are short manuscripts describing new developments or important results of a preliminary nature. These Notes should not exceed 2500 words (where a figure or table counts as 200 words). Following informal review by the Editors, they may be published within a few months of the date of receipt. Style requirements are the same as for regular contributions (see inside back cover).

Ant Colony Algorithms for Two-Impulse Interplanetary Trajectory Optimization

Gianmarco Radice* and German Olmo†
University of Glasgow, Glasgow, G12 8QQ Scotland, United Kingdom

DOI: 10.2514/1.20828

I. Introduction

As operational costs have been increasingly reduced, space systems engineers have been facing the challenging task of maximizing the payload-launch mass ratio while still achieving the primary mission goals. To meet these taxing requirements, during the last two decades, global optimization approaches have been extensively used towards the solution of complex interplanetary trajectory transfers. The methods used include stochastic algorithms such as evolutionary algorithms and simulated annealing, deterministic algorithms, and metamodels such as response surfaces [1–7]. In many cases they have shown the ability to efficiently explore the solution space, providing both unexpected optimal solutions and a number of good initial guesses that have then been further refined through the use of more accurate local optimization techniques. In this paper we present an initial analysis of the effectiveness and usefulness of applying an ant colony algorithm (ACA) to a simple, two-impulse Earth–Mars transfer. Ant colony algorithms have been used to solve hard optimization problems and take inspiration from the observation of the behavior of biological ant colonies [8–11]. This methodology has been applied to the search for families of potentially interesting transfer trajectories from Earth to Mars, in view of future exploration and colonization missions. The ACA has been applied to a simple two-impulse Earth–Mars transfer to assess its suitability and efficiency in identifying local and global optima. We will first present a formulation of the optimization problem, followed by the definition of the optimization method used and the issues encountered in using ACAs for continuous search space applications. It will be shown that the ACA developed here was able to rediscover known solutions as well as identifying new ones.

II. Problem Formulation

As an example of a two-impulse transfer, let us consider a direct transfer from Earth to Mars. We have taken the Mars Express mission as our reference mission. Let us suppose the objective function to be the overall impulsive ΔV at the end of the interplanetary transfer. As a consequence of the mathematical models and methods used for the objective function assessment, the search space is characterized by two design variables: date of departure from Earth, t_0, and transfer time from Earth to Mars, t_t. Upper and lower bounds on the design variables are considered and have been set at [1 January 2003, 31 December 2017] for the transfer time and [100, 600] (days) for the transfer time. The interval of variation has been imposed to include the date of departure of the Mars Express mission (2 June 2003) and seven synodic period of the Earth–Mars system (780 days). As the search space has only two dimensions, a visual representation of the objective function over the whole search space is possible, as shown in Fig. 1. The objective function $\Delta V = f(t_t, t_0)$ is a nonconvex function over the search space considered, mainly due to its quasi-periodical feature on the date of departure values. To further analyze the structure of the objective function, the distribution of the local minima over the whole search domain has been studied. Reeves and Yamada proposed to assess the objective function structure in a flow-shop scheduling environment by firstly identifying as many local minima as possible and then by computing for each local optimum its average distance from all the other local optima, because the global optima for the problem are a priori unknown [12]. Not only does this allow us to identify the best solutions, but also to evaluate the closeness of the local optima to each other, to analyze the structure of the objective function near the global optima by assessing the density and goodness of the nearby local optima, and to identify the presence and features of similar local optima. Because of these attractive features and the important results it led to in the flow-shop scheduling environment, this objective function structure analysis methodology has been applied in this work to space mission design. To generate the local minima, 100 randomly distributed points on the overall search space have been used as starting points for a local search, based on a sequential quadratic programming algorithm. Figure 2 shows the resulting local minima distribution over the search domain. The best solution identified corresponds to a transfer time of 203.541 days and a ΔV of 5678.904 m/s. The interplanetary transfer solution identified here corresponds exactly to the trajectory used by Mars Express. This result can be seen as a confirmation of the validity of the mathematical models and methods used.

III. Ant Colony Algorithm

Ant colony algorithms are inspired by the behavior of natural ant colonies, in the sense that they solve their problems by multiagent cooperation using indirect communication through modifications in the environment. In most ant species, individual ants will deposit a chemical marker, called pheromone that can be sensed by other ants, on the ground while walking [13]. This simple behavior explains why ants are able to adjust to changes in the environment, such as new obstacles interrupting the currently shortest path. When depositing pheromone the ants create a trail that is used to mark the foraging path from the nest to a food source. By sensing the pheromone trails the ants are able to reach food discovered by other ants, and are able to choose the shortest route to it. In analogy to this biological example, ACAs are based on the indirect communication between artificial ants through artificial pheromone trails. The pheromone trails serve as distributed, numerical information used by the ants to find solutions to the problem at hand. During the execution of the ACA, the ants will adapt to the pheromone cues and this will reflect their search experience. The artificial ants iteratively identify a solution for the problem to solve by moving from one state to another one, associated with a better solution, until no further improvements

Received 31 October 2005; revision received 15 March 2006; accepted for publication 20 March 2006.

*Lecturer, Department of Aerospace Engineering. Member AIAA.
†Research Student, Department of Aerospace Engineering.

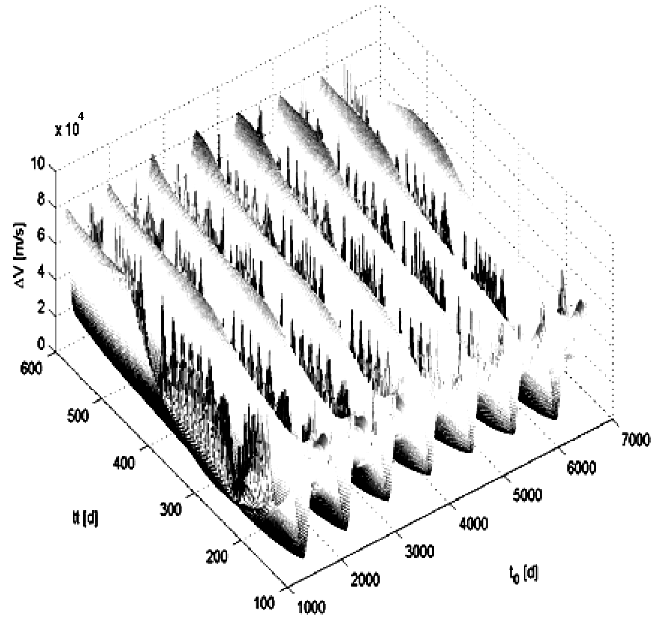

Fig. 1 Total ΔV for a direct impulsive Earth–Mars transfer.

can be achieved. At each step σ, the k ants evaluate a set A_k^σ of possible alternatives to their current state and move to one of these depending on the combination of two values: the goodness η of the move as determined by an appropriate heuristic and the trail level τ of the move indicating the goodness level of previous moves. Paths are updated at each iteration step, thus increasing the level of those that lead to good, or better, solutions, and decreasing all others. All the known ACA applications have the same main structure: starting from a discrete domain, with a set of n nodes to discover the optimum value of a function that is a combination between the nodes and any kind of finite parameters. In other words, all proposed problems to be solved by ACAs are within the field of combinatorial optimization problems. As the problem that we want to solve in this paper, is not within this category, we will have to introduce some modifications to the ACA framework.

IV. Algorithm Design

In traditional combinatorial problems, ants explore the search space by moving from node to node. At each step, the ants make a probabilistic decision choosing the point to move towards among a group of possible options, generally known as neighbors. The ants evaluate the heuristic information and the pheromone trail of each neighbor and then decide their move based on this information. Within a discrete search space, the neighborhood will be a finite set of points and the number of evaluations of the objective function, will depend on the size of the neighborhood but will always be a finite number. However, if the search space is continuous, as is the case of an interplanetary trajectory optimization problem, the number of possible points in the neighborhood of any starting point is infinite. The evaluation of the objective function in each point is therefore obviously impossible. That is why, to make the ACA work for continuous domain problems, some changes in the algorithm have to be done. The idea is to model some kind of discrete structure that allows the continuous and infinite neighborhood to convert itself into a finite group of neighbors. Trying to follow the same main steps than ACA for combinatorial problems, ants are placed inside the search space and made to move along it by selecting points within the group of possible neighbors. These neighbors are a finite set of points selected among the infinite points in the surroundings of each ant. The neighborhood is defined as a nearby region, which encompass the ant, and that contains all the possible neighbors. By selecting n neighbors, the infinite search space has been transformed into a finite one. Now, the ants have a finite group of points to analyze and choose their movement.

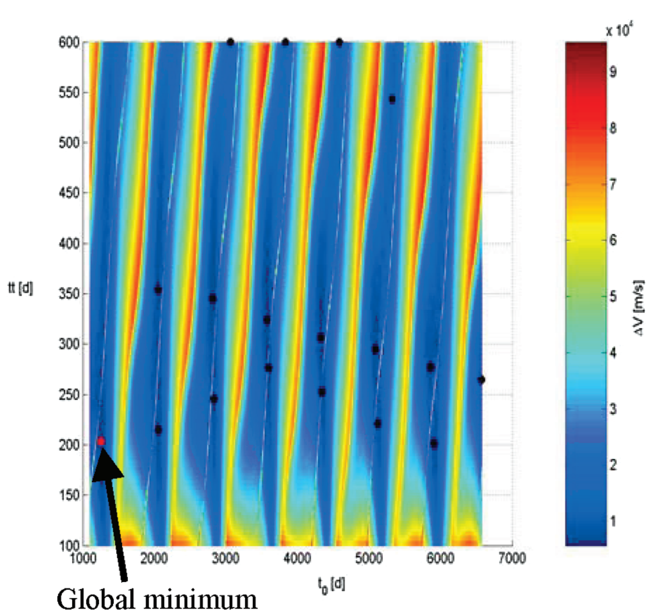

Fig. 2 Local minima distribution.

A. Neighbor Selection

The way in which the neighbors are chosen must be robust enough to ensure that enough points in the search space are sampled and that interesting and promising solutions are not lost. The following issues had to be addressed: ensure that the neighbors chosen covered as many directions as possible; prevent clustering of neighbors, leaving areas of the search space empty; and avoid that successive neighbors are too close to each other. It was decided to use a dual approach to addressing these issues. The neighbors are placed along a circle of radius d with the same angular separation between each other. Note that with only a relatively small number of neighbors, several directions of movements are analyzed. A second batch of neighbors is then selected along a 45 deg arc along the direction of the latest ant movement.

B. Pheromone Trail

As explained previously, ants have the ability of smelling the pheromone trail left by the other ants in the colony. Ants with higher fitness leave a heavier pheromone trail. To reduce the possibility of losing good solutions, a nondeterministic behavior for ants that move following the pheromone trace has been included. The concept hinges on the fact that ants do not always smell the same trail; that is, they can change from time to time the trail to follow in order to search different areas. An ant may smell several trails at any one time. The larger the quantity of pheromone deposited, the more likely it is that an ant will follow that trail. This does not automatically remove the chance that trails with less pheromone are followed. Clearly, however, the trails with higher pheromone levels will be the ones most likely to be selected

C. Colony Reduction

One of the main challenges with such an algorithm is the execution time and its reduction is an important issue. To achieve this we remove those ants that after s steps have not improved their fitness level either through a neighborhood search or through following a pheromone trail.

D. Ant Behavior

At first, when the ant colony has not been reduced, the neighbor distance d is large and decreases linearly at each step k. This allows the inspection of a large area of the search space. The inverse behavior is implemented pheromone trail; that is, the distance at which the ants can smell a trail increases linearly with each step k.

Table 1 ACA results

Run	ΔV, m/s	Iterations
1	5678.90369	1738
2	5678.90366	1794
3	5678.90373	1878
4	5678.90361	1822
5	5678.90356	1694
6	5678.90378	1886
7	5678.90383	1722
8	5678.90363	1983
9	5678.90358	1982
10	5678.90400	1703
σ	1.151×10^{-4}	109.21

This allows the ants to leave their trail and move toward more promising areas of the search space to perform their explorations. Once the local minima have been identified the ants will reduce, following an exponential decrease pattern, the neighbor distance d. This is done to perform a search as thorough as possible in the proximity of all the local minima that have been identified by the ants.

E. Algorithm Considerations

The k ants will move at each time step σ, from their current state i, to a new state j with a probability given by

$$P_{ij}^{k} = \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{J}\tau_{ij}^{\alpha}\eta_{ij}^{\beta}} \tag{1}$$

where η is the heuristic information, τ is the pheromone trail, α and β are two parameters that determine the relative weight of heuristics and pheromone trail, and J is the neighborhood of ant k. The performance of the algorithm is strongly related to the following parameters: the number k of ants in the colony, the number n of neighbor sampling points, and the distance d between the ant and its neighbors. A fourth parameter that does not affect the performance as much as these ones is the distance ds identifying the ant's ability to smell a pheromone trail. The two key indicators of the algorithm's goodness are the number of function evaluations and the effectiveness in identifying the global minimum. It is clear that, for such an algorithm, the two are in contrast with each other; a high number of functions evaluations will lead to better results and vice versa. The selection of the values for k, n, and d, will determine which of the two performance indicators will take precedence. A large number of ants and neighbors and a small distance between them will lead to more accurate results. By reducing the values of these parameters we achieve less function evaluations at the expense of less accurate results.

V. Algorithm Performance

This version of ant colony algorithm was coded in MATLAB and the simulations performed on a Pentium 4 computer with a 2 GHz CPU under Windows XP. To evaluate the performance, the algorithm was initially run 10 times, with 169 ants, having 8 neighbors with search distance $d = 0.02$. The results are shown in

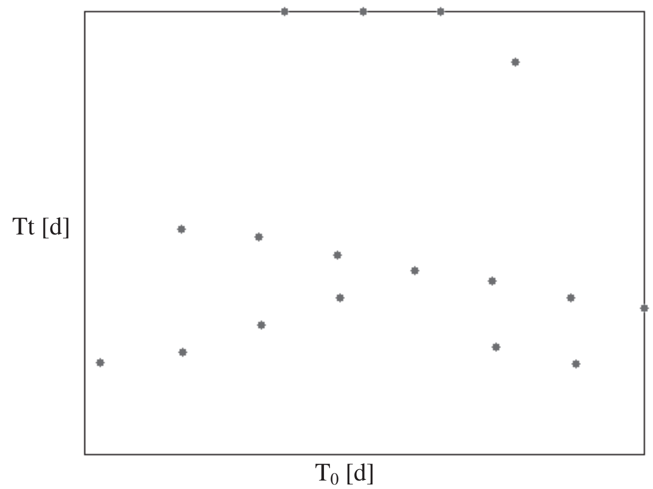

Fig. 3 Minima identified by flow-shop scheduling.

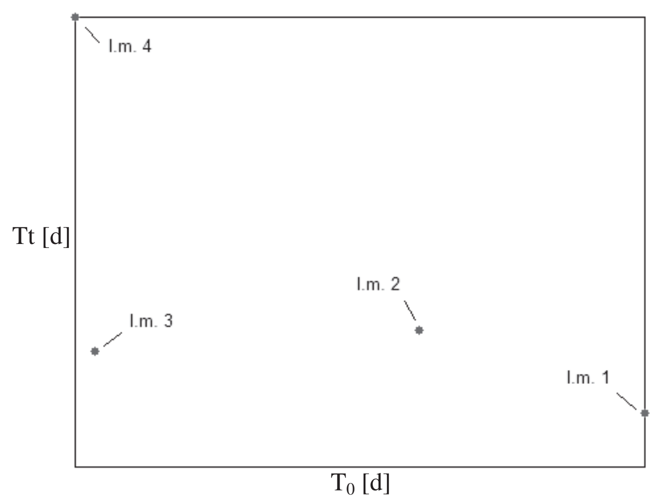

Fig. 4 New minima identified by ACA.

Table 1. Table 2 shows the results from 10 simulations using readily available optimization tools [14]. These include stochastic methods (GAOT, GAOT-shared, GATBX, GATBX-migr, FEP, DE, and ASA), deterministic methods (glbSolve and MCS) and metamodels (rbfsolve). The table shows the best solution achieved by the algorithm in 10 runs, the standard deviation of the solution, and the iterations performed to reach the solution. It can be seen that the performance of the ACA is, for such a simple problem, comparable to that of a deterministic solver, such as MCS, although the iterations necessary to reach the global optimum are more typical of a stochastic solver. What is, however, more interesting to note is that during the simulations four new local minima, not identified through the flow-shop scheduling environment or other optimization algorithms [14], have been found by the ant colony algorithm as shown in Figs. 3 and 4. It is interesting to note that these four local

Table 2 Results for different optimization algorithms

Algorithm	ΔV, m/s)	σ (ΔV)	Iterations	σ (Iterations)
GAOT	5741.524	163.525	1270	345.683
GAOT-shared	6420.207	574.22	590	320.35
GATBX	5740.887	177.082	2322	424.075
GATBX-migr	5679.957	2.191	2650	909.799
FEP	5711.337	95.13	2478	953.829
DE	5986.674	408.679	828	319.692
ASA	6328.291	1330.247	1289	56.555
glbSolve	6406.750	——	565	——
MCS	5678.903	——	640	——
rbfsolve	5684.196	——	953	——

minima correspond to areas of the search space that are highly irregular, and present multiple canals and plateaus. Nevertheless, the ant colony algorithm was able to identify them. The algorithm was also able to identify a local minima (l.m. 1) corresponding to a transfer time of only 161 days, although the ΔV associated with this transfer is over 1 km/s higher than the optimal solution.

VI. Conclusions

In this paper we have attempted to use an ant colony algorithm to solve the problem of a two-impulse Earth–Mars transfer. Because of the continuous nature of the search domain we have had to introduce some changes to the ant colony algorithm, traditionally applied in discrete domains, while still maintaining its biological inspiration. The algorithm successfully identifies the global minimum in all the test cases. In addition to this, the algorithm was also able to identify four new local minima that had not been found by previous analysis. This could lead to identifying different launch dates and launch masses as alternative options to the baseline design. Although the results presented in this paper are promising, there is still a lot of work to be done before an ant colony algorithm can be successfully applied to the optimization of interplanetary trajectories.

References

[1] Betts, J. T., and Orb, S. O., "Optimal Low Thrust Trajectories to the Moon," *SIAM Journal on Applied Dynamical Systems*, Vol. 2, No. 2, 2003, pp. 144–170.

[2] Dachwald, B., "Optimization of Solar Sail Interplanetary Trajectories Using Evolutionary Neurocontrol," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 1, 2004, pp. 66–72.

[3] Gurfil, P., and Kasdin, N. J., "Niching Genetic Algorithms-Based Characterization of Geocentric Orbits in the 3D Elliptic Restricted Three-Body Problem," *Computer Methods in Applied Mechanics and Engineering*, Vol. 191, Nos. 49–50, 2002, pp. 5683–5706.

[4] Hughes, G., and McInnes, C. R., "Solar Sail Hybrid Trajectory Optimization," *Advances in the Astronautical Sciences*, Vol. 109, No. 3, 2001, pp. 2369–2380.

[5] Coverstone-Carroll, V., Hartmann, J. W., and Mason, W. J., "Optimal Multi-Objective Low-Thrust Spacecraft Trajectories," *Computer Methods in Applied Mechanics and Engineering*, Vol. 186, Nos. 2–4, 2000, pp. 387–402.

[6] Rauwolf, G., and Coverstone-Carroll, V., "Near-Optimal Low-Thrust Orbit Transfers Generated by a Genetic Algorithm," *Journal of Spacecraft and Rockets*, Vol. 33, No. 6, 1996, pp. 859–862.

[7] Vasile, M., Summerer, L., and De Pascale, P., "Design of Earth-Mars Transfer Trajectories Using Evolutionary-Branching Technique," *Acta Astronautica*, Vol. 56, No. 8, 2005, pp. 705–720.

[8] Dorigo, M., and Stützle, T., "The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances," IRIDIA, Universite Libre de Bruxelles Technical Report IRIDIA-2000-32, Bruxelles, Belgium, 2000.

[9] Dorigo, M., and Colorni, A., "The Ant System: Optimization by a Colony of Cooperating Agents," *IEEE Transactions on Systems, Man and Cybernetics—Part B, Cybernetics*, Vol. 26, No. 1, 1996, pp. 1–13.

[10] Dorigo, M., and Di Caro, G., "The Ant Colony Optimization Meta-Heuristic," In *New Ideas in Optimization*, edited by D. Corne, M. Dorigo, and F. Glover, McGraw–Hill, London, 1999, pp. 11–32.

[11] Dorigo, M., Di Caro, G., and Gambardella, L. M., "Ant Algorithms for Discrete Optimization," *Artificial Life*, Vol. 5, No. 2, 1999, pp. 137–172.

[12] Reeves, C. R., and Yamada, T., "Genetic Algorithms, Path Re-Linking and the Flowshop Sequencing Problem," *Evolutionary Computation*, Vol. 6, No. 1, 1998, pp. 45–60.

[13] Goss, S., Aron, J., Deneubourg, J. L., and Pasteels, J. M., "Self-Organized Shortcuts in the Argentine Ant," *Naturwissenschaften*, Vol. 76, No. 12, 1989, pp. 579–581.

[14] Di Lizia, P., Radice, G., Vasile, M., and Izzo, D., "On the Solution of Interplanetary Trajectory Problems by Global Optimization Methods," *GO05 International Workshop on Global Optimization*, Almeria, Spain, 2005.