JOURNAL OF SPACECRAFT AND ROCKETS
Vol. 44, No. 3, May–June 2007

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).

Performance Comparison of Stochastic Search Algorithms on the Interplanetary Gravity-Assist Trajectory Problem

Christopher R. Bessette* and David B. Spencer†
Pennsylvania State University, University Park, Pennsylvania 16802

DOI: 10.2514/1.25718

I. Introduction

THE use of evolutionary algorithms (EAs) has seen increased popularity in the field of optimization recently for many problems, including orbit trajectory design problems. The aim of this Note is to perform a comprehensive performance comparison of two EAs: differential evolution (DE) and particle swarm optimization (PSO) on an Earth-to-Jupiter gravity-assist trajectory optimization problem.

Myatt et al. [1] determined that DE outperformed PSO for a slew of Earth–Saturn trajectories, using a variety of gravity assists and deep space maneuvers. Given the uniqueness of individual search spaces, this paper sought to determine if DE reigned supreme on an Earth–Jupiter trajectory using a single gravity assist.

The EAs are serving to optimize, in terms of Δv, a trajectory between Earth and Jupiter which uses a single gravity assist from either Earth or Venus. Considering only the inner planets for gravity assists, Earth and Venus can deliver the most Δv because they have the largest gravitational parameters [2]. The total Δv is composed of up to three separate maneuvers, where Δv_1 is the departure Δv, Δv_2 is the Δv applied at the flyby planet (if needed, a valid solution is $\Delta v_2 = 0$), and Δv_3 is the maneuver needed at the arrival planet. In this example problem, a vehicle is set to leave Earth's heliocentric position sometime after the epoch of 01 January 2005 00:00 UT.

There are three design variables that the EA uses to minimize Δv:
1) TOL (number of days between launch and epoch),
2) TOF_1 (time of flight in leg 1),
3) TOF_2 (time of flight in leg 2).

For the example used throughout this Note, the vehicle is given an 800-Earth day launch window beginning 01 January 2005 00:00 UT. Then TOF_1 is the time the vehicle is in transit to the flyby planet, which has been restricted to either Earth or Venus. Finally, TOF_2 is the vehicle transfer time from the flyby planet to Jupiter. The sum of these three variables becomes the TOA. To determine the transfer arcs, Lambert's solution has been invoked.

II. Application of EAs to the Interplanetary Trajectory Problem

Because this problem is real coded, as opposed to binary coded, only real-coded EAs were considered. As a result, two real-coded algorithms were used in solving this problem. The first, DE, is a heuristic developed by Storn and Price [3] and has shown promise in solving problems with large search spaces. DE is a classical EA in the sense that it functions on the basic premise of a Darwinian natural selection: mutation, crossover, and selection. The second heuristic used in this problem is PSO developed by Kennedy and Eberhart [4], which, unlike DE, does not mimic a Darwinian natural selection; but rather imitates the motion of a swarm of bees, or a school of fish.

This Note serves to identify optimal trajectories between Earth and Jupiter using one gravity assist from either Venus or Earth. Several assumptions were made in the construction of this problem:
1) Two-body dynamics were used.
2) Patched conics were not considered as the gravity of the launch and arrival planets are ignored.
3) Optimal gravitational flybys are assumed, meaning the vehicle "skims" the surface of the flyby planet, thus delivering the maximum Δv gain.

Although the aforementioned assumptions seem restrictive, the main goal of this Note is to identify candidate trajectories. The use of two-body dynamics is a logical first iteration as computational cost is relatively low when compared to the n-body problem. Invoking two-body dynamics can sift out infeasible spaces, thus minimizing the search space in the n-body problem. Furthermore, the rationale behind using an optimal gravity assist is to decrease the design space, thus eliminating further design variables pertaining to the gravity assist.

Lambert's problem was used and requires the following inputs: initial position vector, final position vector, and TOF. Therefore, if the vehicle were leaving Earth with a destination flyby of Venus, the inputs to Lambert's algorithm would be the heliocentric position vector of Earth at launch, the heliocentric position vector of Venus at arrival, and the TOF. Lambert's algorithm then would provide the required Δv for that transfer.

To explain how DE, for example, fits into the interplanetary orbit transfer problem, DE first randomly distributes particles throughout the search space. Then using Lambert's algorithm, the fitness (Δv) of each particle is determined. DE will also mutate solutions, thus perturbing their position in hopes of finding a position which contains a lower fitness. Then DE will mate particles together, thus producing offspring. The offspring's fitness is then determined, and then the "fittest" particles will be selected to continue onto the next generation. Then the same process will occur in the subsequent generation. This continues until either all the particles converge to one point, or the user-defined maximum number of generations has been achieved.

During a "real" interplanetary mission, the vehicle is under the gravitational influence of many bodies that yield very complex gravity fields. In addition, using an n-body gravitational field, the amount of computation required to solve for the trajectories is immense; as a result, two-body dynamics serve as a logical first iteration and are used. Once two-body dynamics are invoked to

Presented as Paper 6306 at the AIAA/AAS Astrodynamics Specialists Conference, Keystone, CO, 20–24 August 2006; received 7 June 2006; revision received 30 October 2006; accepted for publication 28 November 2006.

*Graduate Student, Department of Aerospace Engineering, 229 Hammond Building. AIAA Student Member.
†Associate Professor, Department of Aerospace Engineering, 229 Hammond Building. AIAA Associate Fellow.

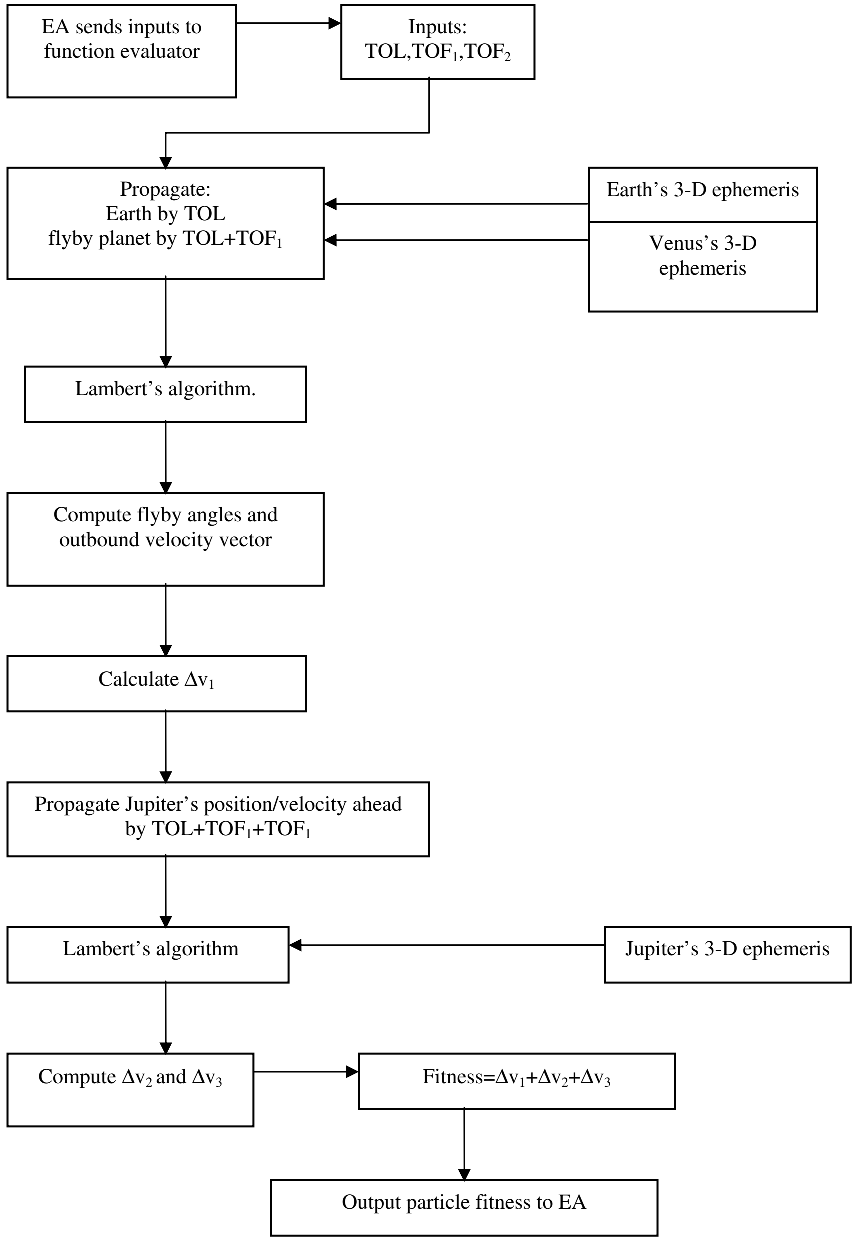

Fig. 1 EA/function evaluator interface.

identify candidate trajectories, those trajectories could be further treated with an n-body gravitational field, and thus realistic optimality of the trajectory could be determined.

To determine the Δv required for a transfer ellipse, a MATLAB script was written. This is the same MATLAB code that the EA calls to determine the fitness of a particular set of input parameters, and it will be referred to as the function evaluator. As a result, for the single flyby case, three decision variables are passed from the EA to the function evaluator: TOL, TOF_1, and TOF_2, with the first TOF being the TOF from Earth to the flyby planet, and the second TOF being the TOF from the second flyby planet to Jupiter. Figure 1 describes the function evaluator algorithm required to determine Δv for a set of input parameters: TOL, TOF_1, and TOF_2. A more detailed description of the function evaluator can be found in [3].

III. Numerical Results

Two possible scenarios, an Earth–Jupiter transfer via either an Earth or Venus flyby, are considered. For the Earth–Earth–Jupiter (EEJ) case, a trajectory was identified with a Δv of 13.43 km/s with a TOA of 1700 days; and the Earth–Venus–Jupiter (EVJ) case required a Δv of 11.82 km/s with a TOA of 1250 days.

EAs typically contain numerous parameters which must be tuned to achieve optimal performance. One major tuning parameter is the population size. Both the EEJ and EVJ scenarios were run with DE and PSO, each using various population sizes. The relative speeds (number of function evaluations versus fitness) of each population size for each algorithm were analyzed; for DE, 25 was deemed the optimal population size, and for PSO, the optimal population size was determined to be 50 [5].

It was hypothesized that PSO would perform the best on the interplanetary problem because it outperformed DE on Rosenbrock's saddle, Rastrigin's function, the LEO–LEO(low Earth orbit) transfer, and the LEO–GEO (geosynchronous Earth orbit) transfer [5]. Figure 2 shows the fitness function (km/s) versus number of function evaluations for this transfer for both DE and PSO for one case for the Venusian flyby scenario.

It is important to note that in Fig. 2, the fitness for the approximately 2000 function evaluations is outside the scale of the y axis. As is evident from this figure, DE arrived at the minimum Δv solution in approximately 2500 fewer function evaluations than PSO. For this comparison, PSO and DE had respective population sizes of 50 and 25. PSO had its cognitive parameter set to 3.0, and its

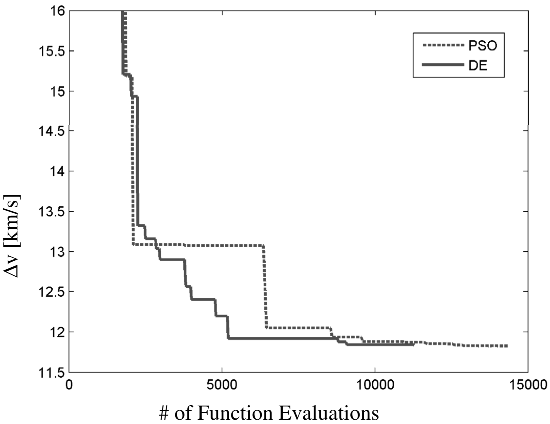

Fig. 2 Fitness versus function evaluations (Earth–Jupiter via Venusian flyby).

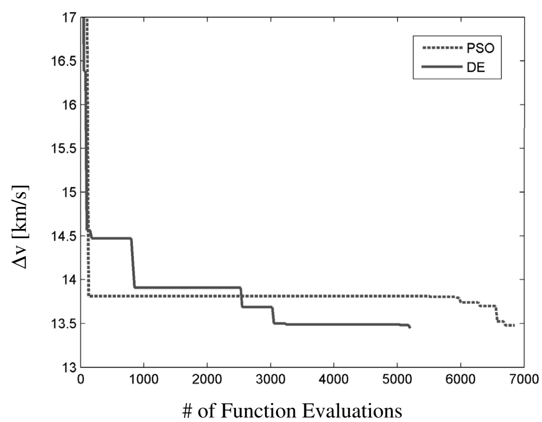

Fig. 3 Fitness versus function evaluations (Earth–Jupiter via Earth flyby).

social parameter set to 1.0. Its constriction factor was also set to 1.0 and its inertia weight was set to 1.0 initially, and then it approached 0.0 as the particles began to converge. DE used a mutation weight of 1.2, and its crossover ratio was set to 0.8. Again, it is important to emphasize that the data in Fig. 2 were created from a single random seed, so a random seed analysis is important to ensure that the data in Fig. 2 are representative of the typical behavior of these heuristics on any given random seed.

The same analysis for the Venus flyby scenario was performed for the Earth flyby case. Figure 3 shows the fitness (km/s) as a function of the number of function evaluations for this case.

As is evident from Fig. 3, DE was able to identify the minimum Δv solution in approximately 1500 fewer function evaluations. However, it is important to note that Fig. 3 was one random seed, and that if each were run numerous times, the outcome could potentially be different. The data for Fig. 3 were created with the same tuning parameters as described for the EVJ scenario.

Because of the computationally intensive nature of this problem, for a random seed analysis, DE and PSO could only be run 10 times each for the Earth swing-by and Venus swing-by cases. A random seed analysis is performed to determine if the data in Figs. 2 and 3 are what one would expect from a given random seed, or if they were aberrations. Table 1 contains the data from the random seed test.

For this test, both DE and PSO were configured as described previously in this section. For both the Venusian and Earth flyby scenarios, both PSO and DE proved to be viable heuristics to handle

Table 1 Random seed results for both Earth and Venus flybys

Case	Average no. of function evaluations	% failed	Median	Standard deviation
Venus flyby				
DE	12,303	0	12,138	1978
PSO	13,280	0	12,975	2062
Earth flyby				
DE	4,576	0	4,835	1023
PSO	7,260	0	7,125	992

this trajectory analysis problem as each was able to arrive at the same minimum Δv solution. Both heuristics converged in a similar number of function evaluations, and they each had a 100% success rate. However, DE was the better of the two heuristics as it did converge quicker than PSO for both scenarios. For the Venusian flyby scenario, DE converged in an average of 8% fewer function evaluations. Similarly, for the Earth flyby case, DE converged in an average of 58% fewer function evaluations. For both PSO and DE, the median was close to the average, thus resulting in a relatively low standard deviation. This shows that both DE and PSO were converging in a similar number of function evaluations for each run. This is an important observation because it illustrates the reliability of DE and PSO, and means that for any random seed, DE converges in about the same number of function evaluations. The same also applies to PSO.

IV. Conclusions

DE and PSO proved robust given their reliability record in the random seed analysis, and they were able to locate the optimum in relatively few function evaluations. Furthermore, both DE and PSO proved to be viable tools for solving the interplanetary trajectory problem for these scenarios, as each was able to identify candidate trajectories without an excessive computational cost. Granted, the use of a single gravity assist in an interplanetary trajectory problem is not considered a daunting task. DE's performance in this case makes it an attractive means to minimize Δv on more involved multiple gravity-assist problems requiring n-body gravity fields.

Although this work uses an epoch of 01 January 2005 00:00 UT, this epoch was chosen arbitrarily. Upon inspection of the final results, it is the author's opinion that this analysis would hold up for any other chosen epoch time.

References

[1] Myatt, D. R., Becerra, V. M., Nasuto, S. J., and Bishop, J. M., "Advanced Global Optimization Tools for Mission Analysis and Design," Final Report of ESA Ariadna ITT AO4532/18138/04/NL/MV, 2004.

[2] Prussing J. E, and Conway, B. A., "*Orbital Mechanics*," Oxford University Press, New York, 1993.

[3] Storn, R., and Price, K., "Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, Vol. 11, No. 4, Dec. 1997, pp. 341–359.

[4] Kennedy, J., and Eberhart, R., "Particle Swarm Optimization," *IEEE International Conference on Neural Networks*, Vol. 4, IEEE, Perth, Australia, 1995, pp. 1942–1948.

[5] Bessette, C. R., "Optimal Interplanetary Trajectories Via Evolutionary Algorithms," M.S. Thesis, Department of Aerospace Engineering, Pennsylvania State University, University Park, PA, 2006.

C. Kluever
Associate Editor