Multi-Objective Evolutionary Algorithms for Low-Thrust Orbit Transfer Optimization

Seungwon Lee, Paul von Allmen, Wolfgang Fink, A. E. Petropoulos, and R. J. Terrile
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive M/S 169-315, Pasadena, CA, USA
+1-818-393-7720
Seungwon.Lee@jpl.nasa.gov

ABSTRACT
We address the problem of optimizing a spacecraft trajectory by using three different multi-objective evolutionary algorithms: i) Non-dominated sorting genetic algorithm, ii) Pareto-based ranking genetic algorithm, and iii) Strength Pareto genetic algorithm. The trajectory of interest is an orbit transfer around a central body when the spacecraft uses a low-thrust propulsion system. We use a Lyapunov feedback control law called the Q-law to create an eligible trajectory, while the Q-law control parameters are selected with the multi-objective algorithms. The optimization goal is to minimize flight time and consumed propellant mass simultaneously. The Pareto fronts (trade-off surface between flight time and propellant mass) produced by these algorithms are evaluated by means of two quantitative metrics: 1) size of the dominated space and 2) coverage of two Pareto fronts. With the two metrics, a hierarchy of algorithms emerged. The non-dominated sorting genetic algorithm and the strength Pareto genetic algorithm are equally effective, and they outperform the Pareto-based ranking genetic algorithm.

Categories and Subject Descriptors
Real-World Applications

Keywords
Multi-objective evolutionary algorithms, Low-thrust orbit transfer, Q-law, Non-dominated sorting genetic algorithm, Pareto-based ranking genetic algorithm, Strength Pareto genetic algorithm, Dominated space, Pareto fronts coverage.

1. INTRODUCTION
Many real-world optimization problems involve multiple competing objectives, which give rise to a set of compromising solutions rather than a single optimal solution. Spacecraft trajectory design is such a multi-objective optimization problem. Optimal trajectories are the ones that minimize both flight time and propellant consumption. Reduction of flight-time often competes with propellant saving. The competition leads to a trade-off between the two resources. The reasonable estimation of the trade-off is the key in spacecraft trajectory design.

The feasibility of the trajectory is bound to the capability of a specific propulsion system. One promising propulsion system for future deep-space missions is electric low-thrust propulsion. In fact, NASA's future space missions Dawn and JIMO will use electric propulsion for inter-planetary cruise and orbital operations. The strength of the electric propulsion is that despite its low thrust levels, the specific momentum transfer per kilogram of propellant is ten to twenty times greater than for chemical propulsion. However, the control of low-thrust spacecraft poses a challenging design problem, particularly for orbit transfers around a central body because it involves a large number of revolutions and thrust arcs along these revolutions.

In an effort to find optimal trajectories, several heuristic control laws have been developed [1,3-5,7,8]. One of the promising control laws is the Q-law, which is based on Lyapunov feedback control [7,8]. The Q-law involves a set of control parameters that are left free for the mission designer to select. The Q-law, with nominal values for the control parameters, provides reasonable estimates of Pareto-optimal solutions, indicating that a suitable Lyapunov function has been found and that optimization of the control parameters should yield near-Pareto-optimal solutions. Indeed, it has been demonstrated that genetic algorithm and simulated annealing optimizations efficiently find the optimal Q-law control parameters for a wide variety of orbit transfers, yielding the estimation of the flight time and propellant mass requirements that are comparable to those obtained with standard, but computationally more demanding, optimization techniques [6,10]. In this paper, we examine the efficiency of several multi-objective evolutionary algorithms for the selection of the Q-law control parameters and thus for the better estimation of the trade-off of the resources in the low-thrust orbit transfer.

2. PROBLEM
2.1 Q-law
The Q-law is a Lyapunov feedback control law developed by Petropoulos in an attempt to provide good initial guesses for optimal trajectories between two arbitrary orbits [7,8]. The Q-law determines when and at what angles to thrust based on the proximity quotient termed Q. The function Q judiciously quantifies the proximity of the osculating orbit to the target orbit. In the Q-law, the central body is modeled as a point mass, and no perturbing forces are considered.

GECCO'05, June 25–29, 2005, Washington, DC, USA.

The Q-law has 13 free control parameters, which mission designers can control. The control parameters affect the efficiency of thrust usage and the geometry (gradient, maxima, minima, saddle points, etc) of the proximity quotient Q. A different efficiency of thrust usage leads to a different length or location of a thrust arc, and a different geometry of Q leads to a different thrust angle or shifts thrust-arc location. Hence, the mission designer can acquire a different trajectory for a different set of the Q-law control parameters. For a detailed discussion of the mechanisms of the Q-law, readers are referred to Refs. [7,8].

2.2 Q-law optimization

For a given set of the control parameters, the output of the Q-law is a series of thrust arcs and thrust angles with a resulting flight time and consumed propellant mass. The desired outcome for the mission designer is the trade-off between flight time and propellant mass and the Pareto-optimal trajectory corresponding to each point on the trade-off surface. Therefore, our optimization problem is to minimize both flight time and required propellant mass while varying the Q-law control parameters. Mathematically, the Q-law optimization problem is expressed as

$$\begin{aligned} \text{minimize } \vec{y} &= \left\{ t(\vec{x}), m(\vec{x}) \right\} \in \mathrm{Y}, \\ \text{where } \vec{x} &= \left\{ x_1, x_2, \cdots, x_{13} \right\} \in \mathrm{X}. \end{aligned} \quad (1)$$

Here, $\vec{x}$ is the Q-law control parameter vector, $\vec{y}$ the objective vector given by the flight time (t) and the consumed propellant mass (m) of the trajectory generated with the given Q-law parameter vector. We call X the decision space and Y the objective space.

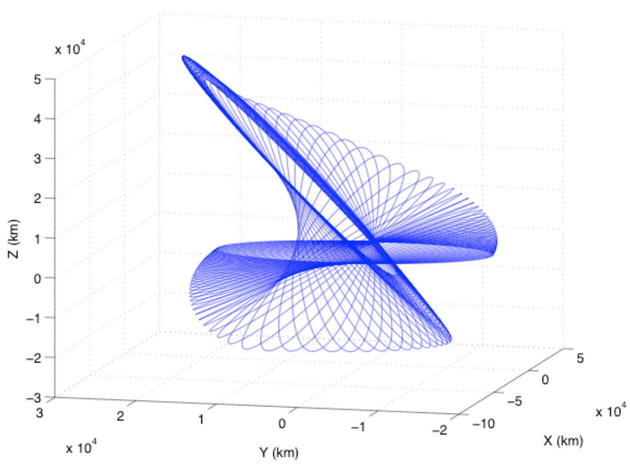

Figure 1. Low-thrust orbit transfer around the Earth between a geostationary-transfer orbit and a retrograde, Molniya-type orbit.

2.3 Low-Thrust Orbit-Transfer

The Q-law optimization is applied to a low-thrust orbit transfer that involves the changes of four of the five orbit elements: the semimajor axis (a), eccentricity (e), inclination (i), the argument of periapsis (ω), and longitude of the ascending node (Ω). More specifically, this is an orbit transfer around the Earth from a geostationary-transfer orbit to a retrograde, Molniya-type orbit, involving a large plane change. We chose this transfer for this study because it is a rather complex orbit transfer, involving changes in most orbit elements. The details of the initial and target orbit are listed in Table 1. The spacecraft initial mass is assumed to be 2000 kg, the thrust 2 N, and the specific impulse 2000 s. A typical minimum-time trajectory for this orbit transfer is shown in Figure 1. This trajectory involves about 80 revolutions around the central body (Earth). The large number of revolutions is common in low-thrust trajectories due to the low level of thrust.

Table 1. Initial and final orbit elements of the orbit transfer around the Earth between a geostationary-transfer orbit and a retrograde, Molniya-type orbit.

Orbit	a (km)	e	i (deg.)	ω (deg.)	Ω (deg.)
Initial	24505.9	0.725	0.06	180	180
Target	26500.0	0.700	116	270	180

3. METHODS

The low-thrust orbit transfer optimization is performed with three different multi-objective evolutionary algorithms: 1) Non-dominated sorting genetic algorithm, 2) Pareto-based ranking genetic algorithm, and 3) Strength Pareto genetic algorithm. The performance of each algorithm is evaluated by means of two quantitative metrics: 1) size of the dominated space and 2) coverage of the two Pareto fronts. We briefly summarize the multi-objective evolutionary algorithms and the performance metrics below. More details can be found in Ref. [11].

3.1 Non-dominated Sorting Genetic Algorithm

The non-dominated sorting genetic algorithm was first implemented by Srinivas and Deb [9]. While it follows the standard genetic algorithm for parent selection and offspring generation, it determines the fitness of the individual using the concept of Pareto dominance as follows. First, the non-dominated individuals in the current population are identified as described in the Appendix. The same fitness value is assigned to all the non-dominated individuals. The individuals are then ignored temporarily, and the rest of the population is processed in the same way to identify a new set of non-dominated individuals. A fitness value that is smaller than the previous one is assigned to all the individuals belonging to the second non-dominated front. This process continues until the whole population is classified into non-dominated fronts with different fitness values. In the original algorithm, the fitness is shared within the decision space. However, we did not apply fitness sharing in our optimization problem because we have not observed any significant improvement in the optimization and the fitness sharing increases the computation time.

3.2 Pareto-based Ranking Genetic Algorithm

The Pareto-based ranking genetic algorithm was proposed by Fonseca and Fleming [2]. Similar to the non-dominated sorting genetic algorithm, this algorithm also uses the concept of Pareto dominance. An individual's rank equals the number of other individuals in the population by which it is dominated. The original algorithm includes fitness sharing in the objective space. However, our optimization omits the fitness sharing.

3.3 Strength Pareto Genetic Algorithm

The strength Pareto genetic algorithm is proposed by Zitzler [11]. It uses the elitism mechanism and the concept of Pareto dominance. It is the elitism mechanism that makes this algorithm quite different from non-dominated sorting or Pareto-based ranking. This algorithm involves an elite group, which is treated differently from the rest of the population. The elite group consists of a subset of non-dominated individuals. The elite group is chosen by clustering the non-dominated set so that the individuals in the elite group are distributed uniformly in the objective space [11].

The fitness assignment procedure is a two-stage process. First, the individuals in the elite group are ranked by a value called strength. The strength of the elite individual is proportional to the number of population members it dominates. The fitness of the individual is equal to its strength. Second, the individuals in the non-elite group are evaluated. The fitness of a non-elite individual is given by the sum of the strengths of all elite individuals who dominate it. Note that fitness is minimized in this algorithm.

3.4 Performance Metrics

The performance assessment of multi-objective optimizers should take at least the following three aspects into account: i) minimal distance to the Pareto-optimal front, ii) adequate (good) distribution, and iii) maximum spread. Various performance metrics to measure the three aspects have been introduced in the literature. We chose the following two measures: I) Size of dominated space and 2) coverage of two Pareto fronts [11]. Zitzler has shown that the two metrics are sufficient to measure the difference in performance between algorithms [11].

3.4.1 Size of the dominated space

The size of the dominated space ***S*** is a measure of how much of the objective space is weakly dominated by a given non-dominated set A. As an example, the size of the dominated space is illustrated in Figure 2. Since our optimization involves the minimization of two objectives, a reasonable maximum value for each objective is chosen to determine the size of the dominated space.

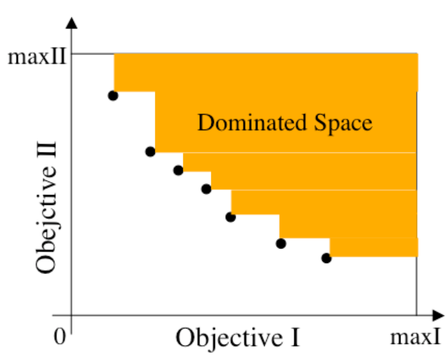

Figure 2. Space dominated (colored in orange) by a given Pareto set when two objectives are minimized.

3.4.2 Coverage of two Pareto fronts

This measure compares two Pareto optimal sets to each other. When two Pareto optimal sets A and B are given, the coverage ***C***(A,B) of the two Pareto fronts maps the ordered pair (A,B) to the interval [0, 1]:

$$C(A,B) = \frac{\left|\{b \in B \mid \exists a \in A : a \succ b\}\right|}{|B|} \tag{2}$$

Therefore, ***C***(A,B) gives the fraction of B dominated by A. For example, ***C***(A,B) =1 means that all individuals in B are dominated by A. The opposite ***C***(A,B)=0 represents the situation that no individual in B is dominated by A. Note that ***C***(A,B) is not necessarily equal to 1-***C***(B,A).

4. APPLICATIONS

4.1 Parameters

For the multi-objective genetic algorithms, the following parameters are used. Each Q-law control parameter is represented by a real-valued genome within a predetermined range. The population size is kept to be 1000. For the strength Pareto genetic algorithm, the size of the elite group is set to be at most 200. The crossover probability is 0.8 and the mutation probability per genome is 0.1. In each generation, 10% of the population is replaced by offsprings. Parents are selected by tournament. Offsprings are created with one-point crossover and random mutation within the predetermined range of each genome. After 200 generations, the genetic evolution is terminated. With these parameter settings, this evolution process evaluates about 19800 different individuals (Q-law control parameter sets).

4.2 Reference Algorithm

As an additional point of reference, random sampling is considered. The random sampling algorithm randomly generates a new individual per generation, according to the rate of crossover and mutation. Hence the number of fitness evaluations is the same as for the other multi-objective algorithms.

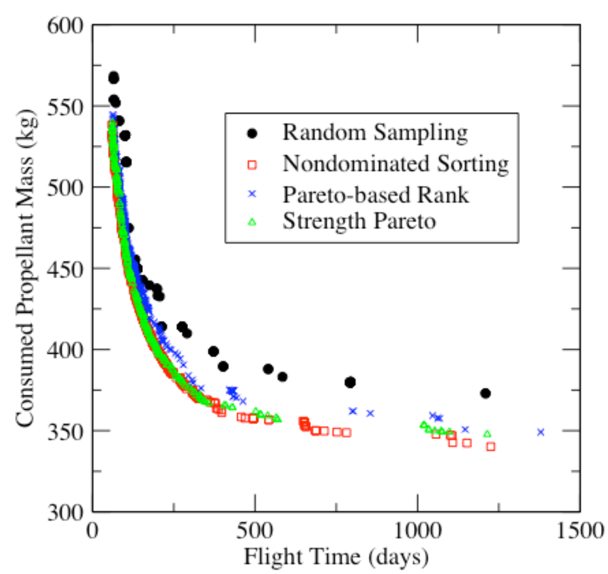

Figure 3. Pareto fronts obtained with the multi-objective algorithms and the random sampling.

4.3 Results

The Pareto fronts obtained with the three multi-objective algorithms and the random sampling are plotted in Figure 3. All of the considered multi-objective algorithms outperform the random sampling. This demonstrates that the genetic algorithms search the design space more efficiently than the random sampling as expected.

The quality of the obtained Pareto fronts are measured and compared according to metrics ***S*** (size of the dominated space) and ***C*** (coverage of two Pareto fronts). Figure 4 shows the size of the dominated space with respect to the number of generations. After 30 generations, the non-dominated sorting outperforms the other multi-objective genetic algorithms. It is interesting to note that up to the tenth generation, the random sampling outperforms the genetic algorithms but is consistently and efficiently outperformed thereafter by the genetic algorithms.

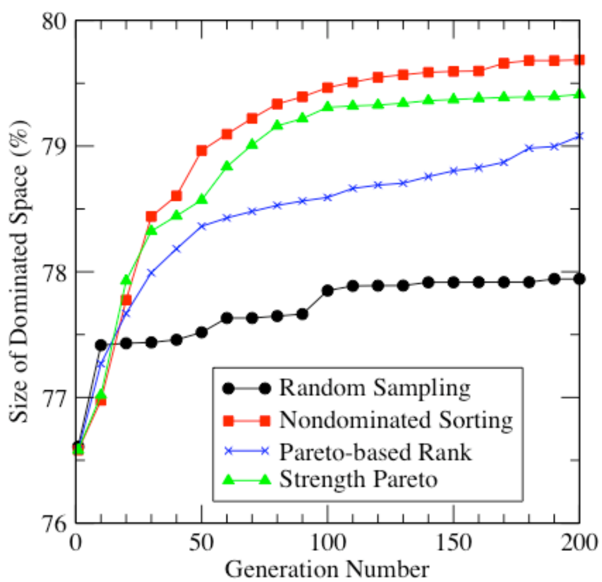

Figure 4. Size of the dominated space vs. generation number for multi-objective genetic algorithms and random sampling.

A more direct comparison between two algorithms is made with the measure of the coverage of two Pareto fronts ***C***(A,B), as listed in Table 2. The Pareto fronts of all the three multi-objective genetic algorithms completely dominate the Pareto front of the random sampling. Among the genetic algorithms, both the non-dominated sorting and strength Pareto genetic algorithms completely dominate the Pareto-based ranking genetic algorithm. Finally, the non-dominated sorting genetic algorithm dominates 76% of the Pareto optimal solutions obtained with the strength Pareto genetic algorithm.

For this test run, both metrics S and C demonstrate that the non-dominated sorting genetic algorithm is superior to the other multi-objective algorithms considered here. To check the robustness of the performance order, two more runs were performed with a different random-number seed. One run leads to the same order of the performance among the algorithms, while the other run switched the order between the non-dominated sorting and the strength Pareto genetic algorithm. In both cases, the Pareto-based ranking genetic algorithm is inferior to other multi-objective genetic algorithms. Considering the results of all the three independent runs, the non-dominated sorting and the strength Pareto genetic algorithms are approximately equivalent, while they clearly outperform the Pareto-based ranking genetic algorithm.

The resulting performance order is different from that obtained by Zitzler for different optimization problems (knapsack problem, traveling salesman problem, continuous test problems) [11]. Zitzler found that the strength Pareto genetic algorithm clearly outperforms the non-dominated sorting algorithm. The disagreement may come from the uniqueness of our problem. Our optimization problem involves the nonuniformity of the objective space. The Pareto-optimal solutions are nonuniformly distributed along the flight-time objective line. The shorter flight time, the more Pareto-optimal solutions there are. This nonuniformity is due to the higher sensitivity of the flight time to the propellant mass in a short flight time zone.

5. CONCLUSIONS

We have applied several multi-objective genetic algorithms to the optimization of low-thrust orbit transfers around the Earth. A Lyapunov feedback control law called the Q-law is used to create an eligible orbit transfer, while the Q-law control parameters are selected with the multi-objective algorithms. Two resources, flight time and propellant mass, are minimized and a trade-off between the two resources is obtained. We have systematically compared the performance of three different genetic algorithms: 1) Non-dominated Sorting Genetic Algorithm (NSGA), 2) Pareto-based Ranking Genetic Algorithm (PRGA), and 3) Strength Pareto Genetic Algorithm (SPGA). Random sampling is also considered as a reference point. Two quantitative performance metrics were used: 1) size of the dominated space and 2) coverage of two Pareto fronts. With these metrics, a hierarchy of the multi-objective genetic algorithms emerged. While NSGA and SPGA are comparable to each other, NSGA and SPGA clearly outperform PRGA. Random sampling performs much worse than all the genetic algorithms.

Table 2. Coverage of two Pareto fronts *C*(A,B) for ordered pairs of algorithms: The listed value *C*(A,B) is the fraction of B dominated by A with algorithm A associated with the corresponding row and algorithm B associated with the corresponding column. The algorithms are Random sampling (RAND), Non-dominated Sorting Genetic Algorithm (NSGA), Pareto-based Ranking Genetic Algorithm (PRGA), and Strength Pareto Genetic Algorithm (SPGA).

***C*(A,B)**	**RAND**	**NSGA**	**PRGA**	**SPGA**
RAND	N/A	0	0	0
NSGA	1	N/A	1	0.76
PRGA	1	0	N/A	0
SPGA	1	0.18	1	N/A

6. ACKNOWLEDGMENTS

This work was performed at the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration. The research was supported by the JPL Research and Technology Development Program.

7. APPENDIX

An individual (a decision vector) $\vec{x}$ is non-dominated by a population set A *iff*

$$\nexists a \in A : \vec{a} \succ \vec{x}, \tag{4}$$

where the condition $\vec{a} \succ \vec{x}$ is defined as

$$\begin{aligned} &\vec{a} \succ \vec{x} \text{ } (\vec{\mathrm{a}} \text{ dominates } \vec{\mathrm{x}}) \\ &\textit{iff} \quad \forall i \in \{1,\cdots,M\}, f_i(\vec{a}) \geq f_i(\vec{x}) \\ &\wedge \exists i \in \{1,\cdots,M\}, f_i(\vec{a}) > f_i(\vec{x}). \end{aligned} \tag{5}$$

Here $f(\vec{a})$ is the objective vector of the decision vector $\vec{a}$, and $f(\vec{x})$ is the objective vector of the decision vector $\vec{x}$.

8. REFERENCES

[1] Chang, D. E., Chichka, D. F., Marsden, J. E., “Lyapunov functions for elliptic orbit transfer,” *AAS/AIAA Astrodynamics Specialist Conference,* AAS Paper 01-441, July 2001.

[2] Fonseca, C. M. and Fleming, P. J., “Genetic algorithms for multiobjective optimization”, *Proc. 5th Int. Conf. Genetic Algorithms*, 416-423, 1993.

[3] Gefert, L. P. and Hack, K. J., “Low-Thrust Control Law Development for Transfer from Low Earth Orbits to High Energy Elliptical Parking Orbits,” *AAS/AIAA Astrodynamics Specialist Conference,* AAS Paper 99-410, Aug. 1999.

[4] Ilgen, M. R., “Low thrust OTV guidance using Liapunov optimal feedback control techniques,” *AAS/AIAA Astrodynamics Specialist Conference,* AAS Paper 93-680, Aug. 1993.

[5] Kluever, C. A., “Simple Guidance Scheme for Low-Thrust Orbit Transfers,” *J. Guidance, Control, and Dynamics*, 21, 6 (Nov. 1998), 1015-1017,

[6] Lee, S., von Allmen, P., Fink W., Petropoulos A. E., and Terrile, R. J., “Design and Optimization of Low-Thrust Orbit Transfers”, will appear in *IEEE Aerospace Conference Proceedings*, Mar. 2005.

[7] Petropoulos, A. E., “Simple Control Laws for Low-Thrust Orbit Transfers,” *AAS/AIAA Astrodynamics Specialist Conference,* AAS Paper 03-630, Aug. 2003.

[8] Petropoulos, A. E., “Low-Thrust Orbit Transfers Using Candidate Lyapunov Functions with a Mechanism for Coasting,” *AAS/AIAA Astrodynamics Specialist Conference,* AAS Paper 04-5089, Aug. 2004.

[9] Srinivas, N. and Deb, K., “Multiobjective optimization using nondominated sorting in genetic algorithms,” *Evolutionary Computation*, 2, 221-248, 1994.

[10] Terrile, R.J., Adami, C., Aghazarian, H., Chau, S.N., Dang, V.T., Ferguson, M.I., Fink, W., Huntsberger, T.L., Klimeck, G., Kordon. M.A., Lee, S., von Allmen, P., Xu, J. “Evolutionary Computation Technologies for Space Systems”, will appear in *IEEE Aerospace Conference Proceedings*, Mar. 2005.

[11] Zitzler, E., “*Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*,” Ph.D. Thesis, Swiss Federal Institute of Technology, Zurich, Switzerland, 1999.