Low-Thrust Mission Trade Studies with Parallel, Evolutionary Computing

Seungwon Lee, Ryan P. Russell, Wolfgang Fink,
Richard J. Terrile, Anastassios E. Petropoulos, and Paul von Allmen
Jet Propulsion Laboratory, 4800 Oak Grove Drive, Pasadena, CA 91109
818-393-7720
Seungwon.Lee@jpl.nasa.gov

Abstract—New mission concepts are increasingly considering the use of ion thrusters for fuel-efficient navigation in deep space. The development of new low-thrust mission concepts requires efficient methods to rapidly determine feasibility and thoroughly explore trade spaces. This paper presents parallel, evolutionary computing methods to access a trade-off between delivered payload mass and required flight time. The developed methods utilize a distributed computing environment in order to speed up computation, and use evolutionary algorithms to find approximately optimal solutions. The methods are coupled with the Primer Vector theory, where a thrust control problem is transformed into a costate control problem and the initial values of the costate vector are optimized. The developed methods are applied to two problems: 1) an orbit transfer around Earth and 2) a transfer between two distance retrograde orbits around Europa. The optimal solutions found with the present methods are comparable to other state-of-the-art trajectory optimizers and to analytical approximations for optimal transfers, while the required computational time is several orders of magnitude shorter than other optimizers thanks to the utilization of the distributed computing environment, the significant reduction of the search space's dimension by the Primer Vector theory, and the efficient and synergistic exploration of the reduced search space by evolutionary computing.

TABLE OF CONTENTS

1. INTRODUCTION

The high fuel efficiency of low-thrust propulsion technologies enables new kinds of missions, since a delivered mass can be increased and/or a trip time can be reduced over chemical propulsion systems. While being highly fuel efficient, the thrust provided by the low-thrust propulsion system is relatively small, typically on the order of one Newton. As a result, any significant maneuver of a spacecraft with the electric propulsion system requires continuous thrust over long periods of time. This makes the low-thrust trajectory optimization more challenging than the chemical-propulsion spacecraft trajectory optimization where only a few impulsive maneuvers need to be optimized.

The development of new low-thrust mission concepts requires methodologies to rapidly determine feasibility and thoroughly explore trade spaces. In particular, a broader assessment of the feasible trade space during early mission design reduces the risk of proceeding with a point-design solution that may be sub-optimal or difficult to implement. The broad assessments also mitigate the risk of missing important design trades that could reduce cost and schedule risk in later mission design phases.

In a prior study, Lee *et al.* demonstrated that parallel, evolutionary computing methods reliably assess a trade-off between flight-time and propellant mass for low-thrust missions. In the current study, the developed method is further applied to a different mission scenario, and the performance of the developed methodologies is analyzed in comparison with state-of-the-art tools and current practices.

2. METHODOLOGIES

In the problem of finding optimal trajectories for low-thrust missions, a common goal is to find the minimum-time, minimum-fuel, or Pareto-optimal trajectories, where the Pareto-optimality means that no other solutions are superior to them in terms of both flight time and fuel consumption. When maximizing the deliverable payload mass is an equally attractive mission objective as minimizing time of flight, the Pareto-optimal solutions that demonstrate the trade-off between flight time and deliverable payload mass are desired. In general, these optimization problems are difficult to solve not only due to continuous thrust over long periods of time but also due to the search for Pareto optimality.

Various methods have been used to solve this optimization problem. A majority of the work has utilized either direct or indirect methods.[1] The direct method approaches the problem by adjusting the control variables iteratively to reduce a performance index such as flight time and propellant mass.[2,3] The continuous control and state variables are often discretized, which results in a nonlinear programming. The indirect method, on the other hand, makes use of the control law that arises when the low-thrust spacecraft problem is formulated using calculus of variations.[4] A co-state vector is introduced and the thrust history is completely determined by the initial values of the co-states. These initial values become the optimization parameters and the only remaining constraint is to hit a specified terminal condition.

For the nonlinear programming problems in the direct method and the indirect method, local gradient-based algorithms such as Newton's method and sequential quadratic programming are popular because they are widely available and proven very effective for many applications. However, the traditional algorithms find locally optimal solutions typically in the vicinity of the initial guesses. Additionally, the algorithms are unstable when the objective function is rugged and the function gradient is discontinuous. The trajectory optimization problem tends to have many locally optimal solutions, which makes it difficult to find the globally optimal solution. Furthermore, the traditional optimization algorithms do not directly handle the multi-objective problem but convert it into a single scalar objective, the so-called weighting method. The resulting solution is highly sensitive to the weighting factors and is a single solution rather than a set of Pareto-optimal solutions. Therefore, the low-thrust orbit transfer problem calls for a more robust, global, and Pareto-optimal optimization algorithm.

This paper presents innovative methods to solve the nonlinear programming problems of the indirect method, so called the Primer Vector theory, for low-thrust mission trade studies involving orbit transfers. The present methods consist of two global-search algorithms: a genetic algorithm and simulated annealing. Neither algorithms require the objective function gradient and are likely to find a globally optimal solution in a rugged search domain, as opposed to gradient-based algorithms. Additionally, the genetic algorithm takes advantage of a population-based search to directly solve the multi-objective optimization problem in a single run. Moreover, the simulated annealing algorithm exploits a highly non-local ensemble search namely "shotgun" mode (described in later section) to efficiently solve the problem in a single parallel run, which exhibits a perfect linear speed-up on a cluster computer due to no communication overhead. Each component of the complementary methods is described below in detail.

Primer Vector Theory

The Primer Vector theory introduces a co-state vector and "indirectly" optimizes the thrust control variables by adjusting the initial values of the co-state vector. This method uses the optimality conditions that arise from calculus of variations to transform the control vector to a function of the initial values of the co-states only. Traditionally, this transformation combined with the Euler-Lagrange boundary conditions, or the so-called transversality conditions, leads to a two-point boundary-value problem, whose solution inherently satisfies the conditions for local optimality. In the current approach, the transversality conditions are ignored, and the initial values of the co-states are iterated to directly optimize the desired objective. This hybrid method is termed indirect however, because the co-states are the control parameters in lieu of the thrust vector itself. The approach as stated is well suited for integration into any general constrained optimization framework, such as the current multi-objective evolutionary computing approaches. This approach is particularly attractive because it avoids one of the problems typically associated with indirect methods: i.e. gradients and transversality conditions must be tediously re-derived each time when an objective or constraint is changed. The transversality conditions are indirectly satisfied by optimizing the initial co-states, and the gradients are not required for the proposed evolutionary optimization methods. Therefore, the problem can be tailored for custom applications with relative ease compared to traditional methods.

Global Optimization Methods

In order to solve the global optimization problems, which appear in the nonlinear programming problem of the Primer Vector theory, the present method uses a genetic algorithm and simulated annealing. The genetic algorithm is inspired by the natural selection and sexual reproduction process of living organisms, and the simulated annealing mimics the thermodynamic process of cooling molten metals. Both methods have mechanisms to escape from local minima in order to find a globally optimal solution. The global search mechanism is a reproduction operator with a stochastic selection mechanism in the case of the genetic algorithm and a mutation operator with the Metropolis algorithm in the case of the simulated annealing.

For the genetic algorithm, the following parameters are typically used: the population size of 1000, the maximum number of generations of 100, the crossover probability of 0.8, the mutation rate per gene of 1/N, where N is the number of genes, the elitist fraction of 30%.

For the simulated annealing, a "shotgun" approach is applied to improve the computational efficiency. In this mode, we start with a random set of values for the co-state parameters and expose them to a fixed (user-defined) number of iterations while the "temperature" of the annealing process is oscillating. If, within the specified iterations, the target orbit is reached with a user-defined accuracy based on the current set of parameters that prescribe an optimal trajectory, then a "solution" is found and stored. Otherwise, the algorithm starts over with another set of random initial values for parameters to optimize. In the results presented here, the number of simulated annealing iterations used for the indirect method optimization was 100.

Pareto Optimization Methods

For the Pareto-optimization, the genetic algorithm handles directly multiple objectives with non-dominated sorting in a single run. The non-dominated sorting uses the concepts of non-dominance and dominance to rank the population composed of candidate solutions. When comparing two solutions, a solution is termed dominated if the solution is inferior to the other solution in all objectives. Otherwise, the solution is termed non-dominated. The non-dominated sorting finds solutions that are non-dominated in comparison with the rest of the candidate solutions in the population. The non-dominated solutions constitute a first Pareto-front and are assigned the best fitness value. The sorting continues with the dominated solutions (i.e., the complement of the non-dominated solutions) by finding the next Pareto-front and assigning a slightly worse fitness value. Since the non-dominated sorting does not involve a weighting process of aggregating the multi-objectives into a single scalar objective function, a careful, educated initial guess of the weighting factors is not needed. The genetic algorithm accompanied by the non-dominated sorting can generate Pareto-optimal solutions in a single synergetic optimization run.

In addition to the nondominated sorting, fitness-sharing mechanisms are applied to the genetic algorithm in order to encourage the wide and uniform spread of the Pareto optimal solutions. The candidate solutions are divided into several groups according to their flight times, and the fitness values of the solutions in the same group are shared in a way such that the sum of the fitness values in every group is the same.

The concepts of non-dominated sorting and fitness sharing do not apply to the simulated annealing approach used here. Instead, a traditional weighting method is used to guide the multi-objective optimization process.

Constraint Handling Methods

The low-thrust trajectory optimization problems involve not only multiple objectives but also multiple constraints such as the boundary condition for a spacecraft final state to meet a given target state. Typically, the constraints are treated with a penalty function as part of the fitness/energy function.[14] The penalty function approach requires a weighting process when combining the penalty function and the objective function into a single scalar fitness function. This approach is used for the simulated annealing application. A different approach named stochastic ranking is used in the genetic algorithm application. The stochastic ranking method strikes a balance between the objectives and the constraints in their contributions to the population ranking process by randomly choosing the ranking criterion between the two. A user-defined parameter determines how probable it is to choose one criterion versus the other.

Parallel Computing Methods

The genetic algorithm uses a population-based search and thus is amenable to parallel computing. When the fitness evaluation is one of the computationally more expensive parts, the parallel computing becomes an ideal choice to reduce the computation time. The fitness evaluation of the candidate solutions in the population is distributed among several processors in the distributed memory system. The evaluation result is sent to the master processor on which the rest of the algorithmic process such as parent selection, offspring creation, and population replacement is executed. The fitness-value passing is the only message passing between the processors in the genetic algorithm run. As a result, the computational overhead due to the parallel computing is marginal.

For the simulated annealing, an “embarrassingly” parallel approach is taken for parallel computing. A perfect linear speed up is demonstrated since each processor performs an independent optimization run without cross-communication (message passing) between processors. We have demonstrated a 1002 CPU run on JPL’s institutional cluster computer (Cosmos, 37th largest super computer in the world to date).

3. MISSION TRADE STUDY RESULTS

The present method is applied to two types of trajectory problems: A) two-body orbit transfer problem and B) restricted three-body orbit transfer problem. The optimization results are presented and compared with solutions found with other state-of-the-art optimizers in terms of solution optimality and computational efficiency.

Orbit Transfer around the Earth

As an example of two-body orbit transfer problems, a low-thrust orbit transfer around the Earth, is considered. The Earth is the only gravitational body in this problem and is approximated as a point mass. The optimization problem is to find Pareto-optimal solutions for the transfer from a low-eccentricity, small orbit to a high-eccentricity, coplanar, larger orbit. The initial orbit has a semimajor axis of 9,222.7 km and an eccentricity of 0.2, and the final orbit has a semimajor axis of 30,000 km and an eccentricity of 0.7. A relatively high thrust magnitude of 9.3 N is used for this transfer problem. The specific impulse of the thrust engine is set to 3,100 s. The initial mass of the spacecraft is 300 kg.

Figure 1 shows the Pareto-optimal solutions found with the present methods. The Pareto-optimal solutions are obtained with various error tolerances between the computed final orbit and the given target orbit, ranging from 1 to 10 percent. As the error tolerance decreases, the obtained solutions converge to “accurate” solutions, which require more flight time and propellant mass as shown in Figure 1.

The present solutions are compared with the solutions found with GA-Q-Law, which is an optimized heuristic control law based on a Lyapunov feedback control law named Q-law and a genetic algorithm. It has been demonstrated that the GA-Q-Law finds nearly Pareto-optimal solutions in a reasonable computation time. The present method efficiently yields solutions comparable to GA-Q-law solutions for a wide range of flight times. Note that GA-Q-law solutions (error<0.03%) have a lower error tolerance than the present solutions. Figure 2 shows four Pareto-optimal trajectories selected among the Pareto-optimal solutions found with the present indirect method. As the flight time increases, several coast (no-thrust) arcs (green dashed lines) are inserted around the apoapsis.

The computational time used to obtain the Pareto optimal solutions is very comparable between GA-Q-Law and the present methods. Both methods utilize parallel computing. With Intel 3.2 GHz Xeon processors, GA-Q-Law used about 5 minutes in wall-clock time on 8 processors, while the present methods used about 5 minutes on 16 processors for each of the 1%, 5%, and 10%-error solution sets.

Distant Retrograde Orbit Transfer around Europa

As a restricted three-body orbit transfer problem, a DRO transfer around Europa, the icy Galilean moon closest to Jupiter, is considered. The gravitational fields of Jupiter and Europa are included as point masses while the gravitational fields of the other moons are excluded. The dynamics of the spacecraft is described in the rotating frame where Europa is at the center, the x-axis points along the Jupiter-Europa line, and the z-axis points along Europa's angular momentum vector with respect to Jupiter. The initial DRO is given by the position vector (0.07518, 0) and the velocity vector (0, -0.14992) in the x-y plane with the unit length of 67,0988 km and the unit time of 48831.6 seconds. Similarly, the final DRO has the position vector (0.03067, 0) and the velocity vector (0, -0.07274). The spacecraft is modeled with the specific impulse of 7,365 s, the thrust magnitude of 4.984 N, and the initial mass of 25,000 kg, which is a spacecraft mass typical for a JIMO (Jupiter Icy Moon Orbiter) Mission.

Figure 3 shows the Pareto-optimal solutions found with the direct and indirect methods. The Pareto-optimal solutions are obtained with error tolerances between the computed final state and the given target state, ranging from 1 to 5 percent. As the error tolerance decreases, the obtained solutions converge to "accurate" solutions. The present optimization results are compared with the solutions found by Mystic, which is a high-fidelity trajectory optimization software package based on the static/dynamic control algorithm. The present method yields results that are comparable to optimal Mystic solutions. Figure 4 shows the variations of the trajectory and control profile for a few selected Pareto-optimal solutions. As the flight time increases, several coast arcs (green dashed lines) are inserted around the *y*-axis.

The computation time used for the Pareto-optimal solutions compares as follows: With Intel 3.2 GHz Xeon processors, Mystic used about 360 minutes on one processor for each of the 0.1%- and 5%-error solution sets, while the present indirect method used about 5 minutes in wall-clock time on 16 processors for the genetic algorithm and 300 minutes on 256 processors for the simulated annealing for each of the 1%- and 5%-error solution sets.

4. PARALLEL COMPUTING PERFORMANCE

The parallel-computing performance of the present methods is analyzed in terms of load balance among processors, message passing overhead, and computational speed-up. For the parallel-computing implementation, the genetic algorithm uses the master-slave architecture, while the simulated annealing uses the embarrassingly parallel architecture. Therefore, load balance and massage-passing overhead should be carefully supervised in the genetic algorithm. On the other hand, they are irrelevant in the simulated annealing, since there is no master processor playing a different role among processors and there is no massage-passing overhead.

Figure 5 shows the load distribution among 8 processors in a genetic-algorithm optimization run. The master node (processor ID =1) has about 10% more load, and slave nodes (processor ID >1) have uniform load. Since the master node plays a different role in the master-slave architecture, the 10% overload in the master node is reasonable. The computational time distribution between the message-passing-related work (MPI functions) and the rest of the work is analyzed for 8 processors in Figure 6. The master node has a different time distribution than other nodes do. The master node spends about 95% of time in MPI functions, while the slave nodes spend only about 8% in MPI function and the rest of time in fitness function evaluation. These distributions are anticipated since the main role of the master node is to distribute the fitness-function evaluation work to slave nodes, to collect the results, and to proceed with ranking and reproduction for the next generation/iteration.

Finally, the computational speedup obtained with the parallel computing is plotted in Figure 7 for both a genetic-algorithm and simulated-annealing optimization runs. The simulated-annealing run demonstrates an almost ideal speedup thanks to no message passing between processors. The genetic-algorithm run shows an 85x speedup with 128 processors. The speedup is reasonable as there is a marginal message passing between master and slave nodes. The speedup of the genetic-algorithm run is directly determined by the ratio of the fitness-evaluation time to the message-passing time. As the ratio increases, the speedup becomes more ideal.

5. COMPARISON WITH CURRENT PRACTICES

To date, there are three low-thrust missions (Deep Space 1, Dawn, JIMO) that have been developed with guidance of available low-thrust design tools. The Deep Space 1 used SEPTOP (Solar Electric Propulsion Trajectory Optimization Program) as the preliminary design tool and NAVTRAJ as the final targeting tool. For Dawn, all the mission design is being done using Mystic. Before being canceled, Prometheus/JIMO used MALTO (Mission Analysis Low Thrust Optimization program) for broad searches and preliminary design of low-thrust gravity-assist heliocentric trajectories to Jupiter. These current tools have also been used on a variety of concept studies and proposals. As an emerging capability, (GA)-Q-Law has been developed to trade studies and preliminary design of planetocentric trajectories, and been integrated into Mystic to assist in generating starting guesses.

The present methods are compared with the current practices in terms of design capability, computational efficiency, and domain expertise requirement. Table 1 highlights several aspects of the direct comparison between the present method and other tools: SEPTOP, Mystic, MALTO, and GA-Q-Law. The design capability of the present method is currently limited to planetocentric and restricted-three-body trajectories, but can be extended to heliocentric trajectories where SEPTOP has demonstrated the applicability of the indirect method (i.e. calculus of variations or Primer Vector theory).

For trade studies, the present method and GA-Q-Law can conduct the study in a single synergistic run while other tools require either multiple independent runs or a single yet parametric run. Although the per-study time varies widely with the type and extent of studies, it can be as long as several months for one broad trade sturdy. By utilizing distributed computing resources, which are abundant nowadays, the prolong per-study time can be reduced to several hours or even several minutes. Both the present method and GA-Q-Law utilize the parallel computing resources.

The complexity of using current low-thrust mission design tools often arises from the extensive requirement of domain expertise. Current tools such as SEPTOP, Mystic, and MALTO require an initial guess to start their optimization processes. The total computational times of these tools greatly depend on the quality of the initial guess. Without an educated guess, the computational time may increase by up to several orders of magnitude or may fail to converge entirely. In contrast, the present method automatically conducts an optimization process with minimal domain expertise for a reasonable bound of each variable. Furthermore, the computational efficiency and solution optimality are less sensitive to the quality of the inputs in the present method.

5. CONCLUSIONS

We have developed a robust and efficient method for low-thrust mission trade studies by applying a genetic algorithm and simulated annealing to the Primer Vector theory. The present method introduces a co-state vector and “indirectly” optimizes the thrust control variables by adjusting the initial values of the co-states. The investigated mission scenarios demonstrate that this method finds nearly Pareto-optimal solutions with a high computational efficiency and minimal guidance from domain expertise. The high computational efficiency is obtained by taking advantage of the utilization of the distributed computing environment, the significant reduction of the search space’s dimension by the Primer Vector theory, and the efficient and synergistic exploration of the reduced search space by evolutionary computing. Future applications of the present method include the optimization of more complex and realistic mission scenarios.

ACKNOWLEDGEMENTS

We thank Christoph Adami for useful discussions about performance analysis for genetic algorithms, and we also thank Robert Tisdale for his support on the parallelization of the simulated annealing. This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The research was supported by JPL’s Research and Technology Development program.

REFERENCES

[1] S. Lee, W. Fink, R. Russell, P. von Allmen, A. E. Petropoulos, and R. J. Terrile, “Evolutionary Computing for Low-Thrust Navigation,” AIAA 2005-6835 Paper, AIAA Space Conference Proceedings, Long Beach, August 30-September 1, 2005.

BIOGRAPHY

Seungwon Lee is a member of the technical staff at the Jet Propulsion Laboratory of California Institute of Technology. Her research interest includes evolutionary computing methods, Astrodynamics, materials and nanostructure simulation, and parallel cluster computing. She received her B.S. and M.S in Physics from the Seoul National University in 1995 and 1997, and her Ph.D. in Physics from the Ohio State University in 2002. Her work is documented in numerous journals and conference proceedings.