

# Optimal Rendezvous Trajectory Using a Hybrid Approach

R. Esmaelzadeh

**Abstract**— Genetic algorithms have gained popularity as effective search procedures for obtaining solutions to traditionally difficult space mission optimization problems. In this paper, a real-coded genetic algorithm is used together with calculus of variations to optimize a trajectory for rendezvous problem. The global search properties of genetic algorithm combine with the local search capabilities of calculus of variations to produce solutions that are superior to those generated with the calculus of variations alone, and these solutions require less user interaction than previously possible. The genetic algorithm is not hampered by ill-behaved gradients and is relatively insensitive to problems with a small radius of convergence. The use of calculus of variations within the genetic algorithm optimization routine increases the precision of the final solution to levels uncommon for a genetic algorithm alone.

**Index Terms**— Trajectory optimization, Genetic algorithms, Hybrid methods, Rendezvous problem.

## 1 INTRODUCTION

TECHNIQUES for optimizing spaceflight trajectory problems have become increasingly important as pressure to reduce the costs of space missions has increased. Both direct and indirect methods, known as “hill-climbing” methods, have been used to optimize space trajectories; however, for some scenarios, convergence to optimal solution is time-consuming, tedious, and sometimes not even possible.

Direct methods that solve for controls to optimize the objective function directly, often via a gradient-based search, suffer from two major drawbacks. First, because search direction is ultimately driven by the local value of the gradient vector, the solution can converge on local, rather than global, minima, resulting in a final solution that is not globally optimal and cannot be further optimized [1]. Second, the optimal solution often has a small radius of convergence, requiring that the guesses for the initial parameters be close to the optimal answer [2].

Indirect methods, such as calculus of variations, obtain optimal results by solving for the costates of a related two point boundary value problem (TPBVP) and not for the controls directly. Although indirect methods are generally more likely to find a true, rather than local, optimum both direct and indirect approaches share many of the same drawbacks, most notably a small radius of convergence [3]. The “hill-climbing” methods exploit all local information in an efficient way, provided that certain conditions are fulfilled and, in particular, that the function to be minimized is “well-conditioned” in the neighborhood of the unique optimum [4]. Such a high level of exploitation requires a lot of local information to be known (gradient and, sometimes, Hessian matrix): the more intensive the exploitation, the stronger the need of specialized information about the function to be minimized. Moreover, if the basic requirements are not satisfied, the reliability of the “hill-climbing” methods is

greatly jeopardized.

Therefore, it is vital to choose initial parameter values intelligently; failure to do so will either dramatically increase the required computation or preclude obtaining a solution entirely. When indirect methods are used, where the optimization parameters are generally not related to the trajectory in an intuitive or straightforward manner, there may not be knowledge of the parameter bounds or their sensitivity. A common strategy to improve initial parameter selection uses previously optimized parameter values from a similar problem as an initial guess [1]. If no closely related solutions exist, initial values are found by optimization of an entire series of intermediate problems relating the new scenario to one with a known solution, a procedure known as homotopy analysis [5].

In recent years many techniques have been suggested for the avoidance of these shortcomings. A survey of these methods can be found in [6]. Evolutionary algorithms (EAs) are the best known. The usefulness of the genetic algorithms (GAs), for solving impulsive trajectories is well documented [7, 8, and 9]. In this paper, the author extends the previous work done on rendezvous trajectory optimization [10]. The purpose of this study was to investigate the GA's effectiveness at determining a near optimal trajectory.

## 2 PROBLEM DEFINITION

The Rendezvous problem is a case that many authors use for demonstrating efficiency of diverse numerical methods, see, e.g., [11, 12]. This problem is summarized as follows:

“For a launch vehicle with a constant-thrust rocket engine,  $a$ , we wish to find the thrust-direction sequence,  $\theta(i)$ , that maximizes final orbital velocity,  $u(t_f)$ , with zero final radial velocity,  $v(t_f)$ , and specified final position  $x_f, y_f$ , for zero initial conditions and given flight time,  $t_f$  (Fig. 1).”

The first order, two dimensional coupled nonlinear differential equations of motion for this problem are

• Faculty of Space Research Institute, Tehran, Iran, E-mail: esmaelzadeh@aut.ac.ir

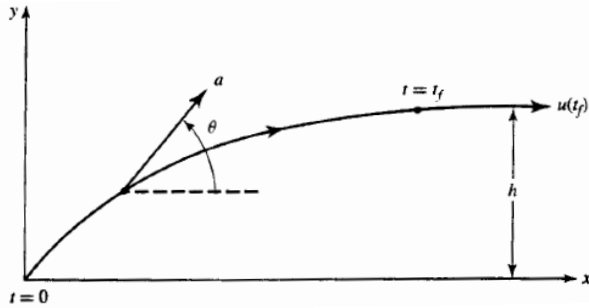


Fig. 1. Nomenclature for rendezvous problem [11].

$$\begin{aligned}\dot{u} &= a \cos \theta \\ \dot{v} &= a \sin \theta - g \\ \dot{x} &= u \\ \dot{y} &= v\end{aligned}\quad (1)$$

where  $(\dot{\phantom{x}})=d(\phantom{x})/dt$ , and  $g$  gravity acceleration. The initial and final conditions for (1) are:

$$\begin{aligned}u(0) &= 0, \quad v(0) = 0, \quad x(0) = 0, \quad y(0) = 0 \\ u(t_f) &= \text{max imum}, \quad v(t_f) = 0, \\ x(t_f) &= x_f, \quad y(t_f) = y_f\end{aligned}$$

If  $\theta$  is held constant for time intervals of length  $\Delta T$ , it is straightforward to show that

$$\begin{aligned}u(i+1) &= u(i) + a\Delta T \cos \theta(i) \\ v(i+1) &= v(i) + \Delta T(a \sin \theta(i) - g) \\ x(i+1) &= x(i) + \Delta T u(i) + 0.5(\Delta T)^2 a \cos \theta(i) \\ y(i+1) &= y(i) + \Delta T v(i) + 0.5(\Delta T)^2 [a \sin \theta(i) - g]\end{aligned}\quad (2)$$

### 3 PROBLEM SOLUTION

#### 3.1 Analytical Solution

At first for studying numerical methods' efficiency, an analytical solution is derived. The augmented terminal cost function is

$$\Phi = u(N) + v_v v(N) + v_y (y(N) - y_f) + v_x (x(N) - x_f) \quad (3)$$

where  $v$  is lagrange multiplier. If we measure time in units of  $t_f$ ,  $(u, v)$  in units of  $at_f$ ,  $(x, y, x_f, y_f)$  in units of  $at_f^2$ , then we can put  $a=1$ ,  $\Delta T=1/N$  in Eq. (3). The  $H(i)$  sequence is then

$$\begin{aligned}H(i) &= \lambda_u(i+1)[u(i) + \Delta T \cos \theta(i)] \\ &+ \lambda_v(i+1)[v(i) + \Delta T(a \sin \theta(i) - g)] \\ &+ \lambda_x(i+1)[x(i) + \Delta T u(i) + 0.5(\Delta T)^2 \cos \theta(i)] \\ &+ \lambda_y(i+1)[y(i) + \Delta T v(i) + 0.5(\Delta T)^2 (\sin \theta(i) - g)]\end{aligned}\quad (4)$$

where  $\lambda$  is lagrange multiplier. The discrete Euler equations are then

$$\begin{aligned}\lambda_u(i) &= H_u(i) = \lambda_u(i+1) + \lambda_x(i+1)\Delta T, \quad \lambda_u(N) = 1 \\ \lambda_v(i) &= H_v(i) = \lambda_v(i+1) + \lambda_y(i+1)\Delta T, \quad \lambda_v(N) = v_v \\ \lambda_x(i) &= H_x(i) = \lambda_x(i+1), \quad \lambda_x(N) = v_x \\ \lambda_y(i) &= H_y(i) = \lambda_y(i+1), \quad \lambda_y(N) = v_y\end{aligned}\quad (5)$$

and the optimality condition  $H_\theta(i)=0$  yields

$$\begin{aligned}\sin \theta(i) [\lambda_u(i+1) + \lambda_x(i+1)\Delta T / 2] \\ + \cos \theta(i) [\lambda_v(i+1) + \lambda_y(i+1)\Delta T / 2] = 0\end{aligned}\quad (6)$$

The adjoint equations are easily solved in this case

$$\begin{aligned}\lambda_x(i) &= v_x, \quad \lambda_y(i) = v_y \\ \lambda_u(i) &= 1 + v_x(1 - i\Delta T), \quad \lambda_v(i) = 1 + v_y(1 - i\Delta T)\end{aligned}\quad (7)$$

Substituting (7) in (6) and using  $N\Delta T=1$  gives bilinear tangent law

$$\tan \theta(i) = \frac{v_v + v_y[1 - \Delta T(i+0.5)]}{1 + v_x[1 - \Delta T(i+0.5)]}$$

(8)

The constants  $v_v$ ,  $v_y$ , and  $v_x$  must be determined to satisfy final conditions. The MATLAB function FSOLVE is used for this propose.

#### 3.2 Genetic algorithms

It is clear that we face a parameter optimization problem. Many different numerical methods have been suggested for solution of these problems, especially in space trajectories applications, that can be found in the Betts's excellent survey [3]. In this paper, a case of genetic algorithms (GAs), known as floating-point or real-coded GA (RGA), is used. The RGAs are a compromise between binary-coded Gas and Evolution strategies [13], since they use most of the classical Genetic Algorithms mechanisms whereas they work directly at the phenotypic level like Evolution Strategies. This RGA generally offers the advantages of being better adapted to numerical optimization for continuous problems, of speeding up the search and of making easier the development of approaches "hybridized" with other methods; but it requires the development of new "genetic-inspired" operators that can be found in [14, 15, and 16].

Whereas traditional methods proceed by deterministically improving an iteration point, GAs use a random "population" of solution candidates, called "individuals," over the entire search space. The features of the best candidates are used for generating new populations, called a "generation," with the intent of producing new and better candidates. The search aims at optimizing a user-defined function (the function to be optimized) called the fitness function. This new generation generally consists of individuals which fit better than the previous ones into the external environment as represented by the fitness function. As the population iterates through successive generations, the individuals will in general tend toward the optimum of the fitness function. This process iterates until one condition in a set of convergence criteria is met.

To generate a new population on the basis of previous one, GA performs three steps [16]: a) it evaluates the fitness score of each individual of the old population, b) it selects individuals on the basis of their fitness score, and c) it recombines these selected individuals using "genetic operators" such as mutation, and crossover, which can be considered as means to change locally the current solutions

and to combine them.

Three important features distinguish the GA approach [16]: a) GA works in parallel on a number of search points and not on a unique solution, which means that the search method is not local in scope but rather global over the search space; b) GA requires from the environment only an objective function measuring the fitness score of each individual and no other information nor assumptions such as derivatives; and c) both selection and recombination steps are performed by using probability rules rather than deterministic ones; this aims at maintaining the global explorative properties of the search.

The convergence of the repeated selection–crossover–mutation procedure to the optimal solution is based on the schema theorem [17], and Markov chain [18]. However, the convergence of GAs is slow, compared to “hill-climbing” methods, when the problem is sufficiently smooth for “hill-climbing” methods to be applicable. This has led to the idea of combining the methods, see e.g., [16]. The GA can be used for generating a starting point for the “hill-climbing” search. Alternatively, the genetic search can be enhanced by performing local “hill-climbing” searches on the members of the population.

The use of GAs to determine optimal space trajectories has only recently gained popularity. The applications range from trajectory planning for launch vehicles to the trajectory design of interplanetary missions [1, 12, 19, 20, 21, and 22].

The RGA used in this study is similar to that described in an orbit transfer problem [22], and simulated with the Genetic Algorithm Toolbox (with some modifications) in MATLAB 8. The number of nodes,  $N$ , for control vector  $\theta(i)$ ,  $i=1\dots N$  through entire trajectory must be defined. As [10] has assumed, we choose  $N=10$ . The RGA task is to find  $\theta(i)$  to maximize fitness function.

For satisfying constraint, using penalty function approach, the fitness function is designed to evaluate the final position and velocity of vehicle at final time as:

$$f = -w_0 u(t_f) + \sum_{i=1}^3 w_i |g_i(x)| \quad (9)$$

$$g(x) = [v(t_f) - x(t_f) - x_f \quad y(t_f) - y_f]$$

The term  $w_i$  are nonnegative penalty factors that are chosen by trial and error to be  $w_0=1$ ,  $w_1=6$ ,  $w_2=2$ , and  $w_3=1.025$ . The basic idea is to assign individuals that have small  $g_i(x)$  a better fitness (or higher  $|u(t_f)|$ ), thereby providing them more opportunity to survive.

By rank method, the raw fitness scores are scaled to values in a range that is suitable for the selection function. This method scales the raw scores based on the rank of each individual instead of its score. The rank of an individual is its position in the sorted scores. The rank of the fittest individual is 1, the next fittest is 2, and so on. Rank fitness scaling removes the effect of the spread of the raw scores.

The other RGA parameters are considered as: stochastic uniform selection with elitism, scattered crossover with 0.8 probability, uniform mutation with 0.1 probability, population size 100, and 50 generations for termination.

The selection function as stochastic uniform lays out a line in which each parent corresponds to a section of the line of length proportional to its scaled value. The algorithm moves along the line in steps of equal size. At each step, the algorithm allocates a parent from the section it lands on. The first step is a uniform random number less than the step size. The scattered crossover creates a random binary vector and selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent, and combines the genes to form a child. The crossover fraction specifies the fraction of the next generation, other than elite children, that are produced by crossover.

The uniform mutation is a two-step process. First, the algorithm selects a fraction of the vector entries of an individual for mutation, where each entry has a probability rate of being mutated. In the second step, the algorithm replaces each selected entry by a random number selected uniformly from the range for that entry.

The operator known as elitism copies the best individual from the previous generation into the new generation if a better individual was not created in the new generation, i.e., elitism was chosen to prevent the current best solution from getting lost. If the individual with the largest value of fitness function in the new generation does not outperform the preceding generation's elite individual, then the old elite individual is copied over the worst performing member of the new generation. The elite count specifies the number of individuals that are guaranteed to survive to the next generation.

The generations, stopping criteria, specifies the maximum number of iterations the genetic algorithm will perform.

The calculations were repeated several times using different seeds to check the repeatability of the optimal parameters. However, a detailed Monte Carlo study to determine their distribution was not performed.

### 3.3 Hybrid approach

GA convergence typically occurred in fewer than 50 generations. After convergence, we have good initial guess for beginning any gradient methods such as DOPC (Discrete Optimization with Constraints) algorithm [11]. DOPC program performs additional calculations to refine the RGA's solution and more precisely define the optimal trajectory. This optimization technique consisted of the RGA and DOPC program working together to find the approximate location of the global minimum, which was further refined by the DOPC program to determine a precise solution. It is not possible to prove that the final solution obtained is a true global minimum [1], but the result can be compared against one obtained with different optimization routines, especially the DOPC algorithm with the assumption that we have very good initial guess, to show that they are superior to or at least equally optimal solution.

### 3.4 Simulation results

With the assumption of  $x_f=0.15$ ,  $y_f=0.2$ , and  $g/a=1/3$ , the optimal trajectories for analytic, GA, and hybrid methods are

shown in Fig. 2. the thrust angle histories obtained from these methods are compared and shown in Fig. 3. This comparison was repeated for state histories in Figs 4 and 5. Table 1 shows final conditions reached by them. a good harmony can be seen from these figures. The hybrid and GA methods reach almost the same position at the same time and satisfy final constraints.

TABLE 1. TRAJECTORY BOUNDARY CONDITIONS

Method	$u_f/at_f$	$v_f/at_f$	$x_f/at_f^2$	$y_f/at_f^2$
Analytic	0.4538	0	0.15	0.2
GA	0.4521	6.7e-4	0.148	0.2
Hybrid	0.4538	0	0.15	0.2

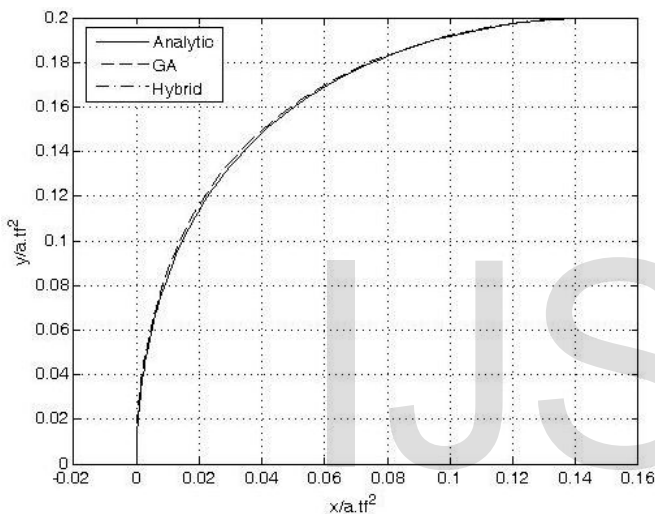


Fig. 2. Trajectory comparisons.

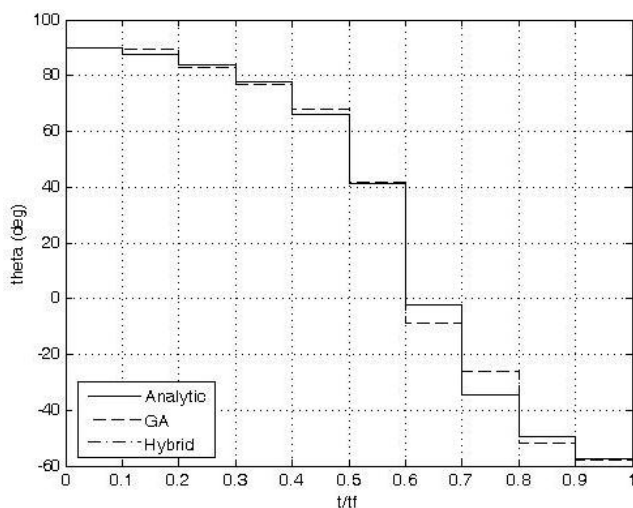


Fig. 3. Control comparisons.

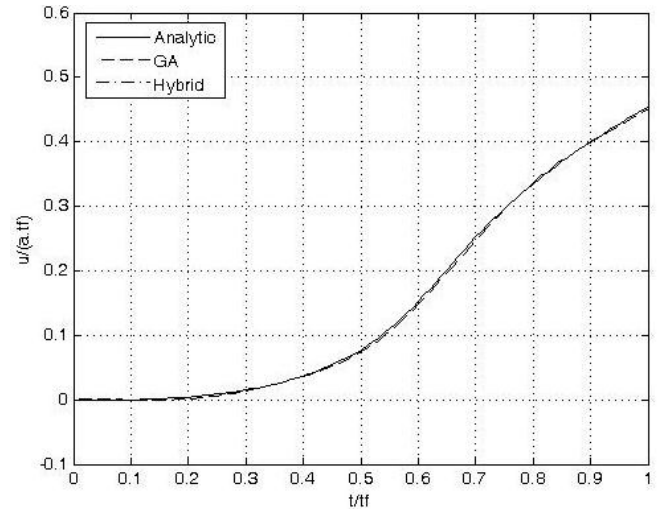


Fig. 4. Orbital velocity comparisons.

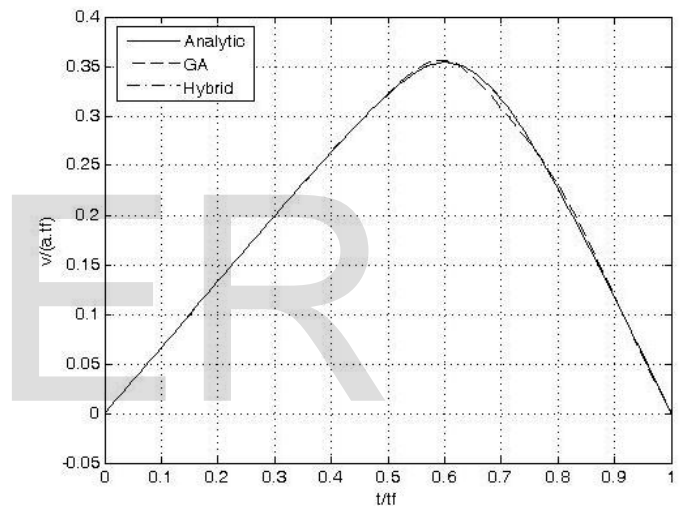


Fig. 5. Radial velocity comparisons.

## 4 CONCLUSION

A real-coded genetic algorithm was used in conjunction with a gradient method (DOPC algorithm) to optimize a rendezvous trajectory. The reliance of the gradient method on earlier solutions and its sensitivity to the quality of the initial guesses were eliminated by relying on the genetic algorithm to search the parameter space to find the location of the globally optimal solution. The DOPC algorithm was used to refine the parameter set found by the RGA, improving the precision of the final answer beyond what would be possible by the use of the RGA alone. to prove that the final solution obtained by hybrid method is a true global minimum, and for investigation of the genetic algorithm solution, the results were compared against one obtained with the analytical method. All methods reached almost the same position at the same time, satisfied final constraints, and had similar control and state histories. Hybrid method proposed here is efficient and robust in achieving global optimal solution when boundary conditions were treated as equality constraints.



## REFERENCES:

- [1] A. Wuerl, T. Carin, and E. Barden, "Genetic Algorithm and Calculus of Variations-Based Trajectory Optimization Technique," *Journal of Spacecraft and Rockets*, Vol. 40, No. 6, pp. 882-888, 2003.
- [2] L. Dewell, and P. Menon, "Low-Thrust Orbit Transfer Optimization Using Genetic Search," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA, Reston, VA, USA, 1999.
- [3] J.T. Betts, "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control and Dynamics*, Vol. 21, No. 2, pp. 193-207, 1998.
- [4] G.S. Beveridge, and R. S. Schechter, *Optimization: Theory and Practice*, New York, McGraw-Hill, 1970.
- [5] R. Bulirsch, F. Montrone, and H.J. Pesche, "Abort Landing in the Presence of a Windshear as a Minimax Optimal Control Problem, Part 2: Multiple Shooting and Homotopy," *Journal of Optimization Theory and Applications*, Vol. 70, No. 2, pp. 223-254, 1991.
- [6] C. R. Reeves, Heuristic search methods: A review, In D. Johnson, F. O'Brien, *Operational Research: Keynote Papers*, Operational Research Society, Birmingham, UK, 1996.
- [7] P.J. Gage, R.D. Braun, and I.M. Kroo, "Interplanetary Trajectory Optimization Using a Genetic Algorithm," *Journal of Astronautical Sciences*, Vol. 43, No. 1, pp. 59-75, 1995.
- [8] M.T. Carter, and S.R. Vadali, "Parameter Optimization Using Adaptive Bound Genetic Algorithms," *AAS Paper 95-140*, 1995.
- [9] E. Pinon, and W.T. Fowler, "Lunar Launch Trajectory Optimization Using a Genetic Algorithm," *AAS Paper 95-142*, 1995.
- [10] R. Esmaelzadeh, A. Naghash, and M. Mortazavi, Global Optimal Solution of Rendezvous Problem Using a Hybrid approach, *The 6<sup>th</sup> International Conference on Simulation, Modeling and Optimization (SMO'06)*, Lisbon, Portugal, September 22-24, 2006.
- [11] A. E. Bryson, *Dynamic Optimization*, Prentice Hall, 1998.
- [12] Y.H. Kim, and D.B. Spencer, "Optimal Spacecraft Rendezvous Using Genetic Algorithms," *Journal of Spacecraft and Rockets*, Vol. 36, No. 6, pp. 859-865, 2002.
- [13] F. Hoffmeister, and T. Back, Genetic algorithms and evolution strategies: similarities and differences, In: H. P. Schefel, R. Manner (eds.) *Parallel Problem Solving from Nature*, Proceedings of 1st Workshop, Dortmund, Germany, 1991.
- [14] K. Deb, *Multi-objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Ltd., 2001.
- [15] M. Gen, and R. Cheng, *Genetic Algorithms & Engineering Optimization*, John Wiley & Sons Inc., 2000.
- [16] J.M. Renders, and S.P. Flasse, "Hybrid Methods Using Genetic Algorithms for Global Optimization," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics* Vol. 26, No. 2, pp. 243-258, 1996.
- [17] N. Ansari, *Computational Intelligence for Optimization*, Kluwer Academic Publishers, 1997.
- [18] N. Seifipour, "A New Online Evolutionary Algorithm, and its Application in Optimization and Control Problems," PhD dissertation, Amirkabir university of technology, Electrical Department, Tehran, 2002.
- [19] V.L. Coverstone-Carroll, J.W. Hartmann, and W.J. Mason, "Optimal Multi-objective Low-thrust Spacecraft Trajectories," *Computer Methods in Applied Mechanics and Engineering*, Vol. 186, pp. 387-402, 2000.
- [20] J.A. Desideri, S. Peigin, and S. Timchenko, "Application of Genetic Algorithms to the Solution of the Space Vehicle Reentry Trajectory Optimization Problem," INRIA Sophia Antipolis, Research Report No. 3843, 1999.
- [21] N. Yokoyama, and S. Suzuki, "Modified Genetic Algorithm for Constrained Trajectory Optimization," *Journal of Guidance, Control and Dynamics*, Vol. 28, No. 1, pp. 139-144, 2005.
- [22] R. Esmaelzadeh, A. Naghash, and M. Mortazavi, Hybrid Trajectory Optimization Using Genetic Algorithms, *Proceedings of the 1st International Conference on Modeling, Simulation and Applied Optimization*, Sharjah, U.A.E., 2005.