LOW-THRUST ORBIT TRANSFER OPTIMIZATION USING GENETIC SEARCH*

Larry D. Dewell† and P.K. Menon‡
Optimal Synthesis Inc., Palo Alto, CA 94306

Most techniques for solving dynamic optimization problems involve a series of gradient computations and one-dimensional searches at some point in the optimization process. A large class of problems, however, does not possess the necessary smoothness properties that such algorithms require for good convergence. Even when smoothness conditions are met, poor initial guesses at the solution often result in convergence to local minima or even a lack of convergence altogether. For such cases, genetic search techniques can be used to obtain a solution. In this paper, trajectory optimization using genetic search methods is illustrated by solving a complex, nonlinear problem involving low-thrust orbit transfer.

Introduction

MUCH work has been done in the past several decades on the solution of complex dynamic optimization problems, particularly in aerospace systems. Solution techniques are generally classified as either "direct" or "indirect". Direct methods solve the optimization problem by directly minimizing the performance index, while indirect methods solve for control functions and parameters which satisfy certain conditions for optimality.[1–3] These techniques generally reduce to solving a multi-point boundary-value problem.[4] The advantage of indirect methods is that they ensure exact satisfaction of optimality conditions, and thus provide guarantees of the local optimality of the solution over a large class of (typically) peicewise-continuous control functions. Direct methods seek to minimize the performance index directly, but only over a restricted, parameterized set of state and control functions. The parameterization allows one to substitute the dynamic optimization problem for a static minimization problem over a finite parameter space, with equality and/or inequality constraints arising from system dynamics and constraints. Several examples of the indirect method exist in the literature.[5,6]

Both direct and indirect techniques rely on some kind of gradient search technique at some point in the solution process. For indirect methods, gradient algorithms are used to satisfy the end conditions of the multi-point boundary value problem. For direct methods, gradient methods are used to determine search directions of lower cost in the finite dimensions the particular parameter space chosen. In order to apply gradient techniques effectively, the system dynamics and performance index must at least have continuous first partial derivatives with respect to state and control. Even when the required smoothness conditions are met, gradient techniques have two significant disadvantages:

- Gradient methods typically have small domains of convergence, resulting in convergence sensitivity to the initial guess. Relaxation techniques can be used to improve the domain of convergence.[7]
- For complicated dynamic optimization problems, gradient methods may prematurely terminate at local inflection points or local minima.

These two drawbacks are significant when not enough is known of the solution structure to form a reasonable initial guess. If the problem can be related to the known solution to another problem related to the original through a homotopy chain,[8] then convergence can be obtained by traversing the chain in small increments. Such a homotopy strategy may not be possible, however, and even when it is, traversal of the homotopy chain can be very time-consuming.

There is thus a need in the dynamic optimization of complex, realistic systems for an optimization technique which does not rely on gradient information, requires no *a priori* knowledge of the solution structure, and where the exact optimality yielded by indirect methods is not essential. The purpose of this paper is to show that genetic search techniques can be powerful tools in the solution of this class of problems. The term "genetic search" describes a set of directed, discontinuous search methods inspired by biological genetics and Darwinian evolution.[9–11] A search problem is cast into the form of a "chromosome" representation; chromosomes are judged based on a "fitness" or performance measure. Chromosomes

*Presented at the 1999 AIAA Guidance, Navigation and Control Conference, Portland, OR

†Research Scientist, 470 San Antonio Road, Suite 200 ; Member, AIAA.

‡President, 470 San Antonio Road, Suite 200 ; Associate Fellow, AIAA

with better fitness values are favored within the population; they are more likely to survive to pass on their traits to offspring. Since genetic search methods are not gradient-based, they are ideal for optimization problems with non-smooth dynamics or performance measures. An extensive set of software tools running in the Matlab® / Simulink® environment has recently been developed which solve optimization problems via genetic search.[12] This software has also been successfully applied to several problems in aerospace guidance and control.[13]

It should be noted, however, that genetic search methods may not be the appropriate solution for certain dynamic optimization problems. If the problem at hand involves smooth system partial derivatives of the dynamics and cost function, known switching structure of any discontinuous controls, and if a reasonable initial guess or the solution to a closely related problem is known, then gradient-based techniques are a better choice. In addition, is should also be noted that while genetic search techniques may yield results where gradient-based techniques fail, the solutions they produce are often sub-optimal.

Genetic Search and Dynamic Optimization

A genetic search is essentially the evolution of a population of chromosomes through genetic operations. To pose the problem as a genetic search, the solution must be represented as a "chromosome". At any stage in the solution process, a "population" or chromosomes exists which contains a set chromosomes, or candidate solutions, to the problem. Each chromosome in the population possesses a "fitness". Members produce offspring via defined genetic operations such as mutation or crossover, thus increasing the number and diversity of the population. The genetic operations are such that the "child" chromosomes inherit some characteristics of the "parent" chromosomes. A "selection" criteria is used to determine which members are used to produce the offspring. Finally, a decimation rule is applied to the population to insure that the population size remains bounded, and that the average fitness of the population improves as the algorithm progresses. The scope of this paper prevents a complete treatment of genetic search algorithms; the reader is instead referred to Ref. 11.

It is helpful in the present context, however, to point out some features of genetic search which are particularly useful in solving dynamic optimization problems.

Chromosome Representation

Genetic searches can only be performed on a problem whose solutions can be expressed as a chromosome which can be operated on by the search algorithm. For dynamic optimization problems, one of the most useful forms for expressing the solution is by an algebraic expression. For example, suppose that one wishes to find the scalar control function $u(t)$ for all $t \in [0, t_f]$ which minimizes a cost function $J(x(\cdot), u(\cdot))$ subject to nonlinear dynamics of the form:

$$\dot{\mathsf{x}}(t) = \mathsf{f}(\mathsf{x}(t), u(t), t) \;\; ; \; \mathsf{x}(0) = \mathsf{x}_0 \tag{1}$$

where $\mathsf{x}(t) \in \mathbb{R}^n$. It is natural to express candidate solutions to this problem as algebraic functions of time. For example, three candidate solutions may include the following:

$$\begin{aligned} u_1(t) &= 1 \\ u_2(t) &= t - \cos(t) \\ u_3(t) &= t^2 - \frac{2e^t}{t+1} \end{aligned}$$

The genetic search package used in this study[12] allows the user to specify chromosomes in a population as string variable algebraic expressions in the Matlab® environment. For example, the above candidate optimal controls could be represented as the following algebraic expressions :

```
u1 = '1'
u2 = 't - cos (t)'
u3 = 't * t - ( ( 2*exp(t) ) / ( t+1 ) )'
```

The above set may constitute the initial population for the genetic search (for a problem of reasonable complexity, the population size would be much larger). By defining an initial population, one is parameterizing, in a sense, the class of control laws which the genetic search optimizes over, and hence, some degree of sub-optimality is implied. For example, all future members of the above population will be formed only from the functions $(1, 2, t, \cos(\cdot), e^{(\cdot)})$ and by the binary operations $(+, -, \times, \div)$. Such a parameterization, however, yields a very rich set of possible functions. For example, after several generations from the above initial population, the following member may be present in the population:

$$u(t) = t^4 - 1 + 2t^2 - t + \cos(e^{2t+1})$$

Genetic Operations

If chromosomes directly represent algebraic expressions, it is necessary that the genetic operation used in the search be such that some sub-expression of a given chromosome be allowed to move across generations, and to be properly combined with sub-expressions from other chromosomes. The appropriate genetic operation which insures this is "crossover", in which random portions of genetic material are selected from each of the two parents and "crossed over" amongst them, producing two offspring. These crossover operations

® Matlab and Simulink are registered trademarks of The Math Works, Inc.

can be performed directly on the string representations of the expressions. Of course, care must be taken in selecting a portion of the genetic material, so that the part selected constitutes a self-contained algebraic expression.

Evaluation of Fitness

The fitness of members of the population must be closely related to the original performance index, J. If chromosomes are represented as algebraic expressions, it is easy numerically to compute $\mathsf{x}(\cdot)$ from equation 1 and, together with $u(\cdot)$ from the chromosome itself, evaluate $J(\mathsf{x}(\cdot), u(\cdot))$.

Algebraic Chromosomes as Series Representations

Although chromosomes as direct algebraic expressions allow a maximum mathematical richness for the dynamic optimization problem, this representation has its disadvantages. First, the crossover of very small amounts of genetic material does not imply correspondingly small differences between the parent and child algebraic functions. Second, the complexity of the chromosome (the number of floating point operations in the associated algebraic expression) grows quickly with successive generations.

For these reasons, it may be desirable to be more restrictive in the parameterization of control functions, while still expressing chromosomes as string algebraic expressions. For example, suppose that the control $u(\cdot)$ was parameterized using Chebychev polynomials of the first kind, as:

$$u(\sigma) = \sum_{i=0}^{N} a_i T_i(\sigma), \tag{2}$$

where $\sigma \in [-1, 1]$ and a one-to-one mapping from t to σ is defined. The Chebychev polynomials

$$T_i(\sigma) = \cos\{i \cos^{-1}(\sigma)\} \tag{3}$$

Under this parameterization, a chromosome in the population could be represented as follows:

```
u = '-1 + 2 * 0.1 + 3 * (2 * 0.1 + 4)'
```

Although this chromosome is treated identically to the more general algebraic expressions above in terms of genetic operations, it is interpreted differently when it is used to evaluate $\mathsf{x}(\cdot)$ in equation 1. For example, this chromosome would represent a Chebychev polynomial with $N = 3$ and with:

$$a_0 = -1$$
$$a_1 = 0.2$$
$$a_2 = 12.6$$

Such a parameterization ensures that small crossover of genetic material result in perturbations of only a single coefficient of the polynomial expansion. Note that the order of the Chebychev polynomial is not fixed during the genetic search. Rather, it is "optimized", together with the coefficients themselves. The order of the Chebychev polynomial, however, grows linearly only with the number of $(+, -)$ operations, with the $(\times, \div)$ operations not increasing the order of the expansion. Of course, this chromosome structure could be used with any other series parameterization for the control; the Chebychev expansion was chosen for the present study of low-thrust orbit transfer.

Enforcing Control Constraints in a Genetic Search

Once a series representation is chosen for the control, it is a simple matter to check for satisfaction of equality or inequality constraints on the control variable $u(\cdot)$. An equality constraint of the form:

$$C(u(t_i), t_i) = 0 \tag{4}$$

translates into an algebraic equation in the expansion coefficients, as in the above Chebychev expansion,

$$\sum_{i=0}^{N} \alpha_i a_i = K \tag{5}$$

In the case of inequality constraints of the form:

$$C(u(t), t) \leq 0 \tag{6}$$

translates into a region $\Im \in \mathbb{R}^N$, such that each admissible $\mathsf{a} = [a_0,\ a_1\ \cdots\ a_N]^T$ must belong to $\Im$. New chromosomes can be evaluated against these two types of constraints, and they can be deleted if the constraint is not satisfied.

Treatment of Discontinuous Control

Just as with the series representation above in which the order of the expansion is not restricted to be fixed during the search, genetic search techniques allow the number of instants of discontinuous control, as well as thier instants in time, to be part of the optimization process. There are many ways this can be accomplished, but one general method which exploits multiple populations is outlined here.

The genetic search tools of Ref. 12 allow for multiple chromosome populations for a given genetic search. For example, two populations could be used to represent separate control functions, say $u_1(\cdot)$ and $u_2(\cdot)$, and a third could represent a switching function $S(\cdot)$, such that if $S(t) < 0$, then $u(t) = u_1(t)$, and if $S(t) > 0$, then $u(t) = u_2(t)$. The number of switching instants between $u_1(\cdot)$ and $u_2(\cdot)$ is determined by the number of positive roots of $S(\cdot)$, and is thus variable during the genetic search.

The Low-Thrust Orbit Transfer Problem

To illustrate the application of genetic search techniques to dynamic optimization, we consider a problem of low-thrust orbit transfer. Suppose that a satellite is in an initial orbit about the Earth, with orbital parameters (ϵ_0, i_0, a_0). It is desired to change this orbit to approach as close as possible to the parameters of a geosynchronous orbit (GEO), given by:

$$\begin{aligned} \epsilon_{\text{des}} &= 0, \\ i_{\text{des}} &= 0, \\ a_{\text{des}} &= 4.21645e7 \text{ meters.} \end{aligned}$$

The orbit transfer will be affected by the continuous burn of a single engine on the spacecraft. The mass rate of change during engine burn is assumed constant. In order for the spacecraft to complete its mission at GEO, it must have a given amount of fuel when the GEO operational orbit is achieved; hence the final time of the orbit transfer is fixed and given by t_f. Figure 1 gives a schematic of the low-thrust orbit transfer problem.

Problem Dynamics and Performance Index

The translational equations of motion of the spacecraft, assuming a perfect inverse-square gravitational field of the Earth and neglecting the mass of the spacecraft relative to the Earth, are given by:[14]

$$\frac{d\mathsf{v}}{dt} = -\frac{\mu_E}{\|\mathsf{r}(t)\|^3} \cdot \mathsf{r}(t) + \frac{1}{m(t)} \cdot \mathsf{T}(t) \qquad (7)$$

where r is the position vector of the spacecraft in an Earth-centered inertial (ECI) frame, v is its time derivative, T is the thrust vector in the inertial frame, and m is the spacecraft mass. The ECI frame has the usual definition, with the x-y plane equal to the equatorial plane, and with the x-axis directed towards Aries. Since these elements will appear later in the performance index, their well-known expressions are repeated here. The semi-major axis a is given by:

$$a(t) = \frac{\mu_E}{2\{\mu_E/\|\mathsf{r}(t)\| - \|\mathsf{v}(t)\|^2/2\}} \qquad (8)$$

Of course, when the thrust T is off, a is a constant of the motion and has physical meaning in terms of the resulting elliptical orbit. During orbit transfer when $\mathsf{T}(t)$ is nonzero, $a(t)$ is the semimajor axis which would result if thrust were turned off at that instant. With this same interpretation, the inclination $i(t)$ is given by:

$$i(t) = \cos^{-1}\left(\frac{\mathsf{h}_z(t)}{\|\mathsf{h}(t)\|}\right) \;;\; \mathsf{h}(t) = \mathsf{r}(t) \times \mathsf{v}(t), \qquad (9)$$

and the eccentricity $\epsilon(t)$ is given by:

$$\epsilon(t) = \sqrt{1 - \frac{\|\mathsf{h}(t)\|^2}{\mu_E a(t)}}. \qquad (10)$$

Based on these parameters, a cost function to measure the efficacy the thrust policy $\mathsf{T}(\cdot)$ in performing the orbit transfer can be written as:

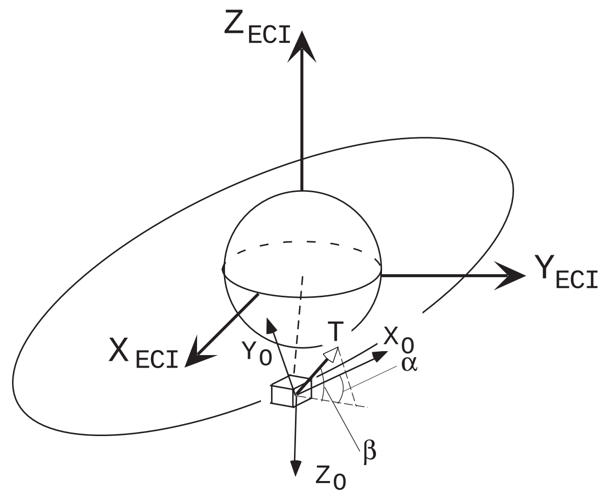

Fig. 2 Definition of frame $\mathcal{O}$ and thrust angles

$$J(\mathsf{T}(\cdot)) = e^{\rho_1|\epsilon(t_f)-\epsilon_{\rm d}|+\rho_2|i(t)-i_{\rm d}|+\rho_3|a(t)-a_{\rm d}|}, \qquad (11)$$

where (ρ_1, ρ_2, ρ_3) are positive real numbers.

Parameterization of Control

For this problem, we suppose that the thrust vector magnitude, and the direction is used to affect the orbit transfer. It is convenient to define the direction of $\mathsf{T}(t)$ in terms of two angles $\alpha(t)$ and $\beta(t)$ defined in an orbit frame $\mathcal{O}$, defined as in Figure 2. The x-axis of frame $\mathcal{O}$ is aligned with $\mathsf{v}(t)$, the y-axis is in the direction of $(\mathsf{r}(t) \times \mathsf{v}(t))$, and the z-axis completes the right-handed triad. As shown in Figure 2, the angle$\alpha(t)$ denotes the angle between the projection of $\mathsf{T}(t)$ onto the x-z plane of frame $\mathcal{O}$, and the angle $\beta(t)$ denotes the angle of $\mathsf{T}(t)$ out of the x-z plane of frame $\mathcal{O}$. Note that there are no explicit physical constraints on the angles.

With these angle definitions, the thrust can be written in the $\mathcal{O}$ frame as:

$$\mathsf{T}_{\mathcal{O}}(\alpha(t), \beta(t)) = T_c \begin{bmatrix} \cos\{\beta(t)\}\cos\{\alpha(t)\} \\ -\sin\{\beta(t)\} \\ \cos\{\beta(t)\}\sin\{\alpha(t)\} \end{bmatrix}, \qquad (12)$$

where T_c is the constant thrust magnitude. The original thrust vector $\mathsf{T}(t)$ can then be written as:

$$\mathsf{T}(t) = A_{\mathcal{I}/\mathcal{O}}(\mathsf{r}(t), \mathsf{v}(t)) \cdot T_{\mathcal{O}}(\alpha(t), \beta(t)) \qquad (13)$$

where the direction cosine matrix $A_{\mathcal{I}/\mathcal{O}}$ is given by:

$$\begin{aligned} A_{\mathcal{I}/\mathcal{O}}(\mathsf{r}(t), \mathsf{v}(t)) &= [\hat{\mathsf{v}}_x \ , \ \hat{\mathsf{v}}_y \ , \ \hat{\mathsf{v}}_z \ ] \qquad (14) \\ \hat{\mathsf{v}}_x(t) &= \frac{\mathsf{v}(t)}{\|\mathsf{v}(t)\|} \\ \hat{\mathsf{v}}_y(t) &= \frac{\mathsf{r}(t) \times \mathsf{v}(t)}{\|\mathsf{r}(t) \times \mathsf{v}(t)\|} \\ \hat{\mathsf{v}}_z(t) &= \hat{\mathsf{v}}_x(t) \times \hat{\mathsf{v}}_y(t) \end{aligned}$$

The angle functions of time $\alpha(\cdot)$ and $\beta(\cdot)$ will be parameterized in terms of finite Chebychev polynomials

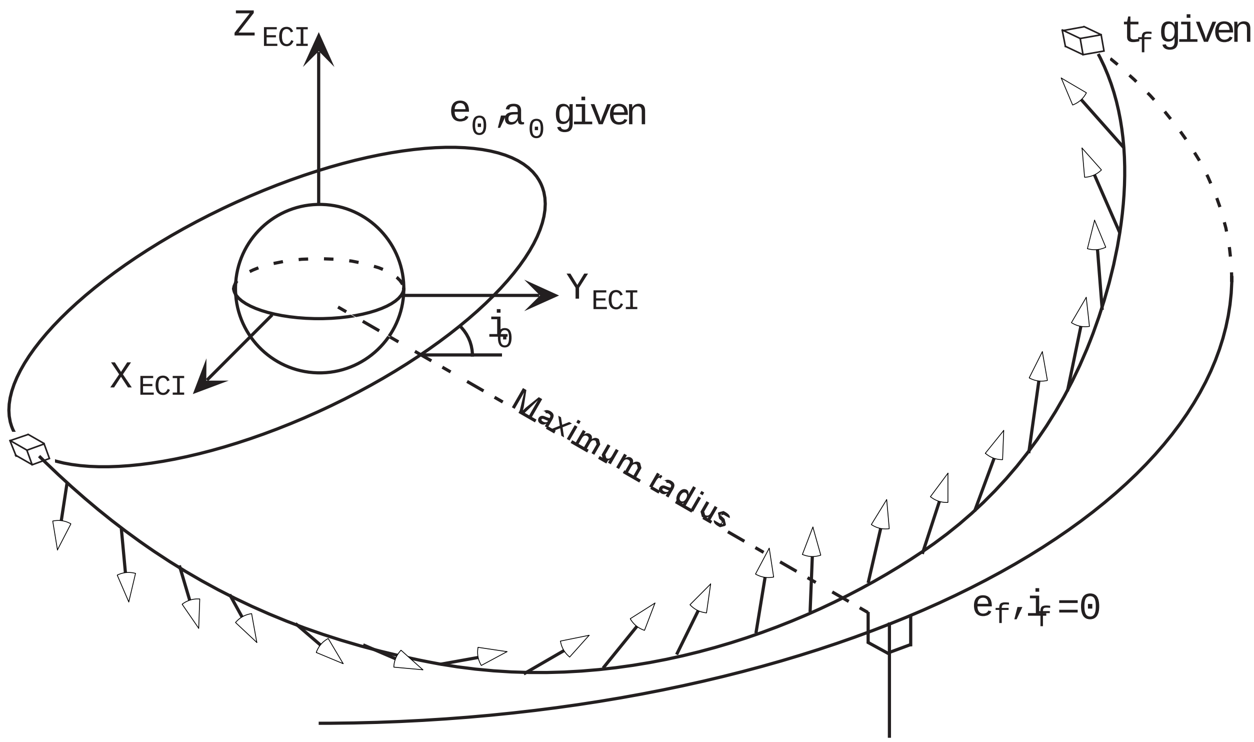

Fig. 1 Schematic of the low-thrust orbit transfer problem

of the first kind, as:

$$\alpha(\sigma) = \sum_{i=0}^{N_\alpha} \alpha_i T_i(\sigma), \qquad (15)$$

$$\beta(\sigma) = \sum_{i=0}^{N_\beta} \beta_i T_i(\sigma), \qquad (16)$$

where $\sigma \in [-1, 1]$. A linear transformation from t to σ was chosen as:

$$\sigma = \frac{2t - t_f}{t_f} \qquad (17)$$

Formulation as Genetic Search

The control variables $\alpha(\cdot)$ and $\beta(\cdot)$ can be represented in the genetic search as two populations of chromosomes, which each chromosome constituting an algebraic expression of Chebychev coefficients, as discussed in Section 2. In the orbit transfer problem, it is reasonable to suspect that at $t = t_f$, the thrust force should be parallel to the velocity vector. This implies that $\alpha(t_f) = \beta(t_f) = 0$. In the interest of further restricting the class of controls searched over by the genetic algorithm, this constraint can be imposed by increasing the order of the Chebychev expansion by one (over that determined by the chromosome structure), and using the terminal equality constraint to compute the additional coefficient. For example, if N_α is the order determined by a given chromosome for $\alpha(\cdot)$, then the actual Chebychev expansion for $\alpha(\cdot)$ is of order $N_\alpha + 1$, and the coefficient $\alpha_{N_\alpha+1}$ is computed from the equation:

$$\sum_{i=0}^{N_\alpha+1} \alpha_i = 0 \qquad (18)$$

It is noted again that under this chromosome formulation, the order of the Chebychev expansion is not fixed, but is optimized during the genetic search.

Results

This orbit transfer problem was considered using the following vehicle parameters and mass properties:

initial vehicle mass = 1534.4kg.
mass loss rate = 9.2e-4 kg./sec.
thrust magnitude = 50 N
final time = 400,000 sec.
perigee altitude of initial orbit = 200 km.
eccentricity of initial orbit = 0.4
inclination of initial orbit = 0.26 rad.

The weighting parameters of the cost function were chosen to be:

$$\rho_1 = 110.0$$
$$\rho_2 = 500.0$$
$$\rho_3 = 1.0\text{e-}9$$

A genetic search was performed for 957 generations. Figure 3 shows a running history of the best fitness (lowest value of the performance index) over the course

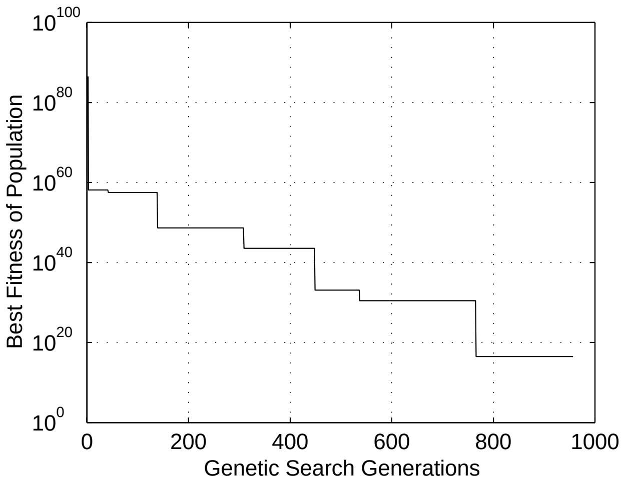

Fig. 3 History of best fitness over the optimization

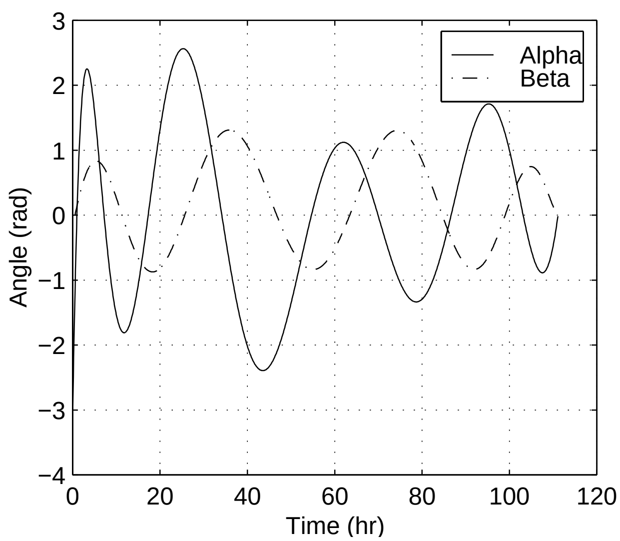

Fig. 4 Thrust angle history for best member after 957 generations

of the optimizations, indicating steady improvement of the performance index by orders of magnitude from a very poor initial fitness. The best members of the populations for $\alpha(t)$ and $\beta(t)$ after the optimization were 11th order Chebychev polynomials. The time history of these angles during the orbit transfer are shown in Figure 4. Figure 5 shows the history of eccentricity $\epsilon(t)$ (given by equation 10) and the history of inclination $i(t)$ (given by equation 9) during the course of the orbit transfer associated with the best members. At the beginning of the orbit transfer, the orbit period is relatively small, as evidenced by the faster oscillations in the orbital parameters. By the final time, however, the orbit has been largely circularized; a larger number of generations would yield even greater circularization of the final orbit. Finally, Figure 6 shows the semi-major axis of the instantaneous orbit. As this graph shows, the thrust profile obtains affects a continuous growth of semi-major axis, a primary objective of the orbit transfer maneuver.

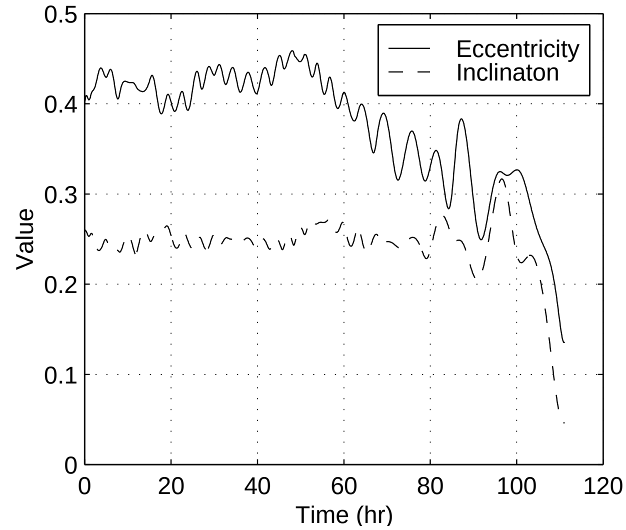

Fig. 5 Eccentricity and Inclination for best member

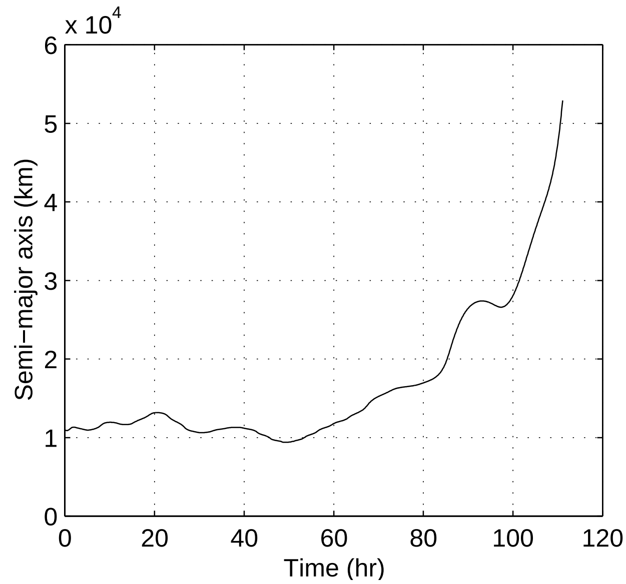

Fig. 6 Semi-major axis history for best member

Summary

In this paper, dynamic optimization problems have been examined from the point of view of genetic search. Genetic search techniques may be the appropriate solution choice for optimization problems which are nonsmooth, or when not enough is known about the optimal solution to generate a sufficiently good initial guess. The genetic search technique was illustrated here by solving a nonsmooth, nonlinear problem involving low-thrust orbit transfer. The results obtained illustrate general convergence to the minimum, although a more lengthy genetic search should be conducted to achieve greater optimality of the result.

References

[1]Bryson, A., *Dynamic Optimization*, Addison Wesley Longman, Inc., Menlo Park, CA, 1999.

[2]Maurer, H. and Gillessen, W., "Application of Multiple Shooting to the Numerical Solution of Optimal Control Problems with Bounded State Variables," *Computing*, Vol. 15, 1975, pp. 105–126.

[3]Miele, A., "Recent Advances in Gradient Algorithms for Optimal Control Problems," *J. of Optimization Theory and Applications*, Vol. 17, No. 5/6, 1975.

[4]Oberle, H. and Grimm, W., "BNDSCO - A Program for the Numerical Solution of Optimal Control Problems," Tech. Rep. 515-89/22, Institute for Flight Systems Dynamics, German Aerospace Research Establishment DLR: Oberfaffenhofen, 1989.

[5]Vlassenbroeck, J., "A Chebyshev Polynomial Method for Optimal Control with State Constraints," *Automatica*, Vol. 24, No. 4, 1988, pp. 499–506.

[6]von Stryk, O. and Bulirsch, R., "Direct and Indirect Methods for Trajectory Optimization," *Annals of Operations Research*, Vol. 37, 1992, pp. 357–373.

[7]Deuflhard, P., "A Relaxation Strategy for the Modified Newton Method," *Optimization and Optimal Control*, No. 477 in Lecture Notes in Mathematics, Springer Verlag, Berlin, 1975, pp. 59–73.

[8]Bulirsch, R., Montrone, F., and Pesche, H., "Abort Landing in the Presence of a Windshear as a Minimax Optimal Control Problem, Part 2 : Multiple Shooting and Homotopy," *J. of Optimization Theory and Applications*, Vol. 70, 1991, pp. 223–254.

[9]Holland, J., *Adaptation in Natural and Artificial Systems*, The MIT Press, Cambridge, MA, 1993.

[10]Fogel, D., *Evolutionary Computation*, IEEE Press, Piscataway, NJ, 1995.

[11]Koza, J., *Genetic Programming*, The MIT Press, Cambridge, MA, 1992.

[12]Optimal Synthesis Inc., *Genetic Search Toolbox User's Manual*, Optimal Synthesis Inc., Palo Alto, CA, 1999.

[13]Crawford, L., Cheng, V., and Menon, P., "Synthesis of Flight Vehicle Guidance and Control Laws using Genetic Search Methods," *AIAA Guidance, Navigation and Control Conference*, Portland, OR, 1999.

[14]Sidi, M., *Spacecraft Dynamics and Control : A Practical Engineering Approach*, Cambridge University Press, 1997.