which after substitution of Eqs. (19) and (20) and the identity in Eq. (4) leads to

$$\dot{H}_I^T K = H_I^T R\left[\frac{\partial}{\partial \mu}(R^T K\dot{\mu}) - R^T \dot{K}\right] = 0 \qquad (22)$$

Because the matrix $K(\mu)$ is singular only for certain Euler angle coordinates, whereas the relation in Eq. (22) is true for all values of the Euler angles, we can conclude $\dot{H}_I = 0$. This implies that the angular momentum of the rigid body is a conserved quantity.

Conclusion

The conservation of angular momentum of a free rigid body is a well-known fact. When the rigid body is described in terms of its Eulerian motion, however, this conservation law cannot be readily established. We present a new identity involving Euler angles and their rates. This identity provides a Noetherian perspective of free Eulerian motion and readily establishes the well-known fact that the angular momentum is conserved.

Acknowledgment

The second author would like to gratefully acknowledge the support provided by the National Science Foundation, NSF Grant CMS-9410157.

References

[1]Lanczos, C., *The Variational Principles of Mechanics*, 4th ed., Dover, New York, 1970.

[2]Meirovitch, L., *Methods of Analytical Dynamics*, McGraw–Hill, New York, 1988, Chap. 4.

Near-Optimal Low-Thrust Trajectories via Micro-Genetic Algorithms

Victoria Coverstone-Carroll*
University of Illinois at Urbana–Champaign, Urbana, Illinois 61801

Introduction

GENETIC algorithms (GAs) are robust parameter optimization techniques based on the Darwinian concept of natural selection. Genetic algorithms have been successfully employed to determine both impulsive[1–3] and low-thrust orbit transfers.[4] The simplest form of a GA uses the three basic operators of reproduction, crossover, and mutation. Populations typically range from 30 to 200 individuals.[5] Here, the effectiveness of using micro-genetic algorithms (μGAs) to determine near-optimal low-thrust trajectories is investigated. Micro-GAs are GAs with populations typically fewer than 20 individuals. Schemes for using μGAs have been proposed[6] and have been shown to be more effective (i.e., have fewer function evaluations) than larger population GAs.[7,8]

The GA approach is extremely powerful at solving unconstrained optimization problems; however, the extension to constrained optimization remains a research issue. Two methods for handling constraints are studied. One technique enforces constraints through equality constraints appended to the objective function,[4] whereas the second approach constrains the problem via inequality constraints.[9]

Low-Thrust Problem Formulation

A control profile that minimizes propellant consumption while satisfying specified boundary conditions is to be determined. The planar dynamics in polar coordinates (r, θ) are given in Eqs. (1–4).[10] The radial velocity is denoted by u, tangential velocity v, gravitational constant of the attracting body μ, initial mass m_0, propellant consumption rate $\dot{m}$, and time t:

$$\dot{r} = u \qquad (1)$$

$$\dot{\theta} = v/r \qquad (2)$$

$$\dot{u} = \frac{v^2}{r} - \frac{\mu}{r^2} + \frac{T\sin(\phi)}{m_0 - \dot{m}t} \qquad (3)$$

$$\dot{v} = -\frac{uv}{r} + \frac{T\cos(\phi)}{m_0 - \dot{m}t} \qquad (4)$$

The control profile is given by the thrust magnitude T and thrust direction ϕ. Because a GA is a parameter optimization technique, these continuous functions must be represented by a finite string of numbers. Therefore, the entire trajectory is broken into discrete segments. The GA then determines the thrust direction and magnitude at the beginning of each segment. The control is assumed constant over a segment, and the equations of motion are integrated. The total time required to complete the trajectory is fixed. In addition, the propulsion system contains a constant power source such as a nuclear electric engine and produces a constant thrust magnitude. The optimal thrust magnitude profile for these types of propulsion system typically consists of periods of burn arcs at maximum thrust and periods of coasting. Therefore, the thrust magnitude is modeled by a switching function modulating between on and off. The propellant consumption rate is supplied by Eq. (5), where the specific impulse of the propellant is denoted by I_{sp} and the Earth's gravity by g:

$$\dot{m} = T/gI_{sp} \qquad (5)$$

GA Characteristics

Genetic algorithms were developed by John Holland and his students in the 1970s,[11] and a detailed description can be found in Ref. 5. For the GA used in this study, the parameters in the thrust profile (T and ϕ) are converted into a finite length string of binary characters. The engines are either on or off so that a single bit representation is sufficient for the thrust magnitude representation. The thrust direction resides between 0 and 360 deg. A 5-bit string is used, providing a resolution of 11.25 deg.

The simple GA used in this study is similar to that described in a previous low-thrust optimization study.[4] Tournament selection, single point crossover, and mutation were used. The μGA used tournament selection and uniform crossover, and these two operations continue until the population converges, typically every four to five generations. Convergence is considered achieved when fewer than 5% of the total number of bits of each individual are different from the best series of bits for that generation. Once this occurs, the best solution is copied over to the next generation, and the remaining population is again randomly recreated and the procedure repeats itself. Uniform crossover was chosen because it has been shown to be more robust than single point crossover for μGAs when treating problems with multiple local minimizing solutions.[8] No mutation is used in μGAs, because with the rapid convergence cycles mutations do not have time to evolve before a new random population is introduced.

Constraint Introduction

Genetic algorithms are well suited to solving unconstrained optimization problems of the form: Determine the vector $\boldsymbol{x}$ of dimension n to maximize the scalar function $f(\boldsymbol{x})$. However, when m equality constraints on $\boldsymbol{x}$ are introduced, a uniform procedure for posing the problem to GAs is not available. Here, two methods are investigated. In the following discussions, the m-dimensional vector of constraints is represented by Eq. (6):

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0} \qquad (6)$$

Received April 16, 1996; revision received May 30, 1996; accepted for publication July 2, 1996.

*Assistant Professor, Department of Aeronautical and Astronautical Engineering, 306 Talbot Laboratory, 104 S. Wright Street. Member AIAA.

Equality Constraints

The first technique involves maximizing the objective function shown in Eq. (7). The terms w_i are nonnegative penalty factors. Based on the results of a previous low-thrust study,[4] it is extremely unlikely that all $g_i(\boldsymbol{x})$ will reach zero before the GA is assumed converged:

$$\boldsymbol{J} = w_0 f(\boldsymbol{x}) + \frac{1}{\sum_{i=1}^{m} w_i g_i^2(\boldsymbol{x})} \tag{7}$$

The basic idea is to assign individuals that have small $g_i(\boldsymbol{x})$ a large fitness (objective function), thereby providing them more of an opportunity to survive. Solutions that do not come close to satisfying the constraints are quickly weeded out of the population with this method.

Inequality Constraints

The second technique was recently introduced by Seywald and Kumar.[9] In Ref. 9, the crossover operator is proven to be inefficient when applied to objective functions with equality constraints enforced through the previously mentioned method, especially if the individuals are heavily penalized. A new method was presented that treats equality constraints as inequality constraints. This technique helps prevent premature loss of genetic diversity. The method will be generally stated and then applied specifically to the low-thrust spacecraft trajectory problem.

Consider slightly perturbing the equality constraints in Eq. (6):

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{e} \tag{8}$$

The vector $\boldsymbol{e}$ is dimension m and consists of small positive quantities. Next envision determining the vector $\boldsymbol{x}$ of dimension n to maximize the scalar function $f(\boldsymbol{x})$ subject to the constraints in Eq. (8). Denote this solution $\boldsymbol{x}^*(\boldsymbol{e})$. A new constraint function $\boldsymbol{h}(\boldsymbol{x})$ is defined, the m components of which depend on whether the optimal cost $f[\boldsymbol{x}^*(\boldsymbol{e})]$ increases or decreases when the ith component of $\boldsymbol{e}$ is increased. The exact definition of $h_i(\boldsymbol{x})$ is

$$h_i(\boldsymbol{x}) = -g_i(\boldsymbol{x}) \quad \text{if} \quad f[\boldsymbol{x}^*(\boldsymbol{e})] > f[\boldsymbol{x}^*(\boldsymbol{0})]$$

$$h_i(\boldsymbol{x}) = +g_i(\boldsymbol{x}) \quad \text{if} \quad f[\boldsymbol{x}^*(\boldsymbol{e})] < f[\boldsymbol{x}^*(\boldsymbol{0})] \tag{9}$$

In this definition $f[\boldsymbol{x}^*(\boldsymbol{0})]$ denotes the scalar function $f(\boldsymbol{x})$ when $\boldsymbol{e}$ is zero and the constraints are exactly satisfied. The m $h_i(\boldsymbol{x})$ in Eq. (9) provide gradient information on how $f(\boldsymbol{x})$ changes due to a small positive change in the ith constraint. Engineering insight can provide the information needed to determine the vector $\boldsymbol{h}$. In the cases where intuition does not provide the essential information, the unconstrained problem can be solved first and this information used to form $\boldsymbol{h}$. Once the $\boldsymbol{h}$ vector is determined, an additional vector $\boldsymbol{j}$ of dimension m is defined. The ith component of $\boldsymbol{j}$ is given by

$$j_i = \min(0, h_i) \tag{10}$$

Finally, a new objective function is constructed where the terms k_i are nonnegative weighting terms:

$$J = k_0 f(\boldsymbol{x}) + \sum_{i=1}^{m} k_i j_i(\boldsymbol{x}) \tag{11}$$

Appending the vector $\boldsymbol{j}$ in Eq. (11) has the same effect as determining the vector $\boldsymbol{x}$ of dimension n to maximize the scalar function $f(\boldsymbol{x})$ subject to the inequality constraints in Eq. (12):

$$\boldsymbol{h}(\boldsymbol{x}) \geq \boldsymbol{0} \tag{12}$$

Orbit Transfer Problem

A trajectory from Earth to Mars' orbit provides the test case. Circular orbits for both Earth and Mars are assumed, and canonical units are used.[12] The spacecraft begins in the Earth's orbit. The initial conditions for the problem are

$$r(0) = 1.0\,\text{DU}, \qquad \theta(0) = 0.0\ \text{rad}$$
$$u(0) = 0.0\,\text{DU/TU}, \qquad v(0) = 1.0\,\text{DU/TU} \tag{13}$$

Here, DU represents a distance unit and TU a time unit in canonical units. Likewise, the boundary conditions $r(t_f)$, $\theta(t_f)$, $u(t_f)$, and $v(t_f)$ represent the desired position and velocity of the spacecraft when the transfer is complete. The boundary conditions corresponding to a circular Mars orbit are

$$r(t_f) = 1.524\,\text{DU}, \qquad \theta(t_f) = \text{free}$$
$$u(t_f) = 0.0\,\text{DU/TU}, \qquad v(t_f) = 0.81\,\text{DU/TU} \tag{14}$$

Note that the final angular displacement $\theta(t_f)$ is not specified, because an orbit-to-orbit transfer is being modeled. The total flight time is 200 days. The spacecraft's thrust magnitude is 3.787 N, initial mass is 4545.5 kg, and mass consumption rate when the engines are thrusting is $6.787e^{-5}$ kg/s.

The equality constraint objective is to maximize Eq. (7) with the scalar function $f(\boldsymbol{x})$ given by $f(\boldsymbol{x}) = m_{\text{ga}}(t_f)$ subject to the constraint vector in Eq. (15). The subscript ga on a variable indicates the final value of the parameter determined by the GA, and the subscript f indicates the desired final value. The weighting constants used were chosen by trial and error to be $w_0 = 2.5 \times 10^{-3}$, $w_1 = 1 \times 10^4$, $w_2 = 1 \times 10^4$, and $w_3 = 1 \times 10^4$:

$$\boldsymbol{g}^T(\boldsymbol{x}) = [r_{\text{ga}}(t_f) - r_f \quad u_{\text{ga}}(t_f) - u_f \quad v_{\text{ga}}(t_f) - v_f]^T = \boldsymbol{0}^T \tag{15}$$

The inequality constraint objective is to maximize Eq. (11) using the scalar function $f(\boldsymbol{x})$ given by $f(\boldsymbol{x}) = m_{\text{ga}}(t_f)$ subject to the inequality constraint vector in Eq. (16). The weighting constants k_i are all set to the value 1:

$$\boldsymbol{h}^T(\boldsymbol{x}) = \{r_{\text{ga}}(t_f) - r_f \quad -[u_{\text{ga}}(t_f) - u_f] \quad v_{\text{ga}}(t_f) - v_f\}^T \geq \boldsymbol{0}^T \tag{16}$$

Results: Equality Constraints Method

Figure 1 displays the objective function for the best individual in the population as a function of the number of objective function evaluations. Note that the vertical scale on the left should be used. Three curves are shown: a simple GA with a population of 150 individuals and two μGAs with populations of 5 and 15. The simple GA crossover probability was set at 0.8 and mutation probability at 0.005. The crossover probability for the μGA was set at 0.5. Figure 1 shows that the μGA with a population of 15 evolves faster than the other two GAs but both μGAs stagnate to a suboptimal region. The

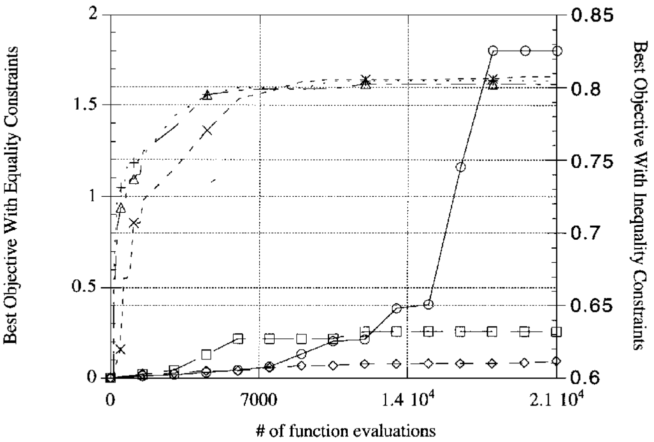

Fig. 1 Objective function vs function evaluations: ○, simple GA, eq. con.; □, μ-GA, population = 15, eq. con.; ◇, μ-GA, population = 5, eq. con.; ×, simple GA, ineq. con.; +, μ-GA, population = 15, ineq. con.; △, μ-GA, population = 5, ineq. con.

simple GA is slow to evolve but after 120 generations (18,000 function evaluations) reaches a near-optimal solution. The boundary conditions for the best solution in the simple GA are $r(t_f) = 1.519$ DU, $\theta(t_f) = 147.18$ deg, $u(t_f) = 0.005$ DU/TU, and $v(t_f) = 0.807$ DU/TU and for the μGA with a population of 15 are $r(t_f) = 1.514$ DU, $\theta(t_f) = 162.59$ deg, $u(t_f) = 0.206$ DU/TU, and $v(t_f) = 0.805$ DU/TU. The μGAs' radial position and tangential velocity are close to the desired values given in Eq. (14) but the radial velocity would make the final orbit eccentric.

Results: Inequality Constraints Method

Figure 1 also displays the objective function for the best individual in the population using inequality constraints. The vertical scale on the right should be used. It is seen that, given a sufficient number of generations, all three GAs converged to a near-optimal solution. However, the μGAs converged to the near-optimal region more quickly than did the simple GA. The performance of all GAs became comparable after approximately 7500 function evaluations. The boundary conditions for best solution in the μGA with a population of 15 after 6000 are $r(t_f) = 1.509$ DU, $\theta(t_f) = 153.41$ deg, $u(t_f) = 0.042$ DU/TU, and $v(t_f) = 0.808$ DU/TU and after 12,000 function evaluations are $r(t_f) = 1.524$ DU, $\theta(t_f) = 152.59$ deg, $u(t_f) = 0.051$ DU/TU, and $v(t_f) = 0.807$ DU/TU.

Orbital rendezvous trajectories have also been solved using μGAs with inequality constraints but are not shown here. In all cases the μGA provides an extremely fast approach to a near-optimal trajectory.

Conclusions

The use of μGAs to determine near-optimal low-thrust trajectories was explored. Micro-GAs were inefficient at achieving near-optimal solutions when boundary conditions were treated as equality constraints. However, when boundary conditions were cast as inequality constraints, μGAs showed faster convergence than did simple GAs to a near-optimal region.

Acknowledgment

The author would like to acknowledge David Carroll for providing the GA driver used in this study and for valuable discussions pertaining to its use.

References

[1]Gage, P. J., Braun, R. D., and Kroo, I. M., "Interplanetary Trajectory Optimization Using a Genetic Algorithm," *Journal of the Astronautical Sciences*, Vol. 43, No. 1, 1995, pp. 59–75.

[2]Carter, M. T., and Vadali, S. R., "Parameter Optimization Using Adaptive Bound Genetic Algorithms," American Astronomical Society, Spaceflight Mechanics Conf., AAS Paper 95-140, Albuquerque, NM, Feb. 1995.

[3]Pinon, E., III, and Fowler, W. T., "Lunar Launch Trajectory Optimization Using a Genetic Algorithm," American Astronomical Society, Spaceflight Mechanics Conf., AAS Paper 95-142, Albuquerque, NM, Feb. 1995.

[4]Rauwolf, G., and Coverstone-Carroll, V., "Near-Optimal Low-Thrust Orbit Transfers Generated by a Genetic Algorithm," *Proceedings of the 18th Southeastern Conference on Theoretical and Applied Mechanics*, Univ. of Alabama, Tuscaloosa, AL, 1996, pp. 431–442.

[5]Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison–Wesley, Reading, MA, 1989, pp. 112, 113.

[6]Goldberg, D. E., "Sizing Populations for Serial and Parallel Genetic Algorithms," *Proceedings of Third International Conference on Genetic Algorithms* (Fairfax, VA), Clearinghouse for Genetic Algorithms, Dept. of Engineering Mechanics, Univ. of Alabama, Tuscaloosa, AL, 1989, pp. 70–79.

[7]Krishnakumar, K., "Micro-Genetic Algorithms for Stationary and Non-Stationary Function Optimization," Society of Photo-Optical Instrumentation Engineers, Conf. on Intelligent Control and Adaptive Systems, SPIE Paper 1196-32, Philadelphia, PA, Nov. 1989.

[8]Carroll, D. L., "Genetic Algorithms and Optimizing Chemical Oxygen-Iodine Lasers," *Proceedings of the 18th Southeastern Conference on Theoretical and Applied Mechanics*, Univ. of Alabama, Tuscaloosa, AL, 1996, pp. 411–430.

[9]Seywald, H., and Kumar, R.,"Genetic Algorithms and Equality Constrained Optimization Problems," *Journal of Guidance, Control, and Dynamics* (submitted for publication).

[10]Bryson, A. E., and Ho, Y. C., *Applied Optimal Control*, Hemisphere, New York, 1975, pp. 66–69.

[11]Holland, J., *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA, 1992, Chap. 1.

[12]Bate, R. R., Mueller, D. D., and White, J. E., *Fundamentals of Astrodynamics*, Dover, New York, 1971, pp. 40, 41.

Conjecture About Orthogonal Functions

Larry Silverberg*
North Carolina State University, Raleigh, North Carolina 27695-7910

Introduction

I FIRST came upon this result in my work in the area of control of distributed systems. I was studying the manner in which the natural modes of vibration of simple beams are altered by attaching to them concentrated spring and damping elements. The result has since been distilled into an unproven theorem, presently being called the orthogonal function conjecture.

Orthogonal function conjecture: Let $\phi_r(x)$ $(r = 1, 2, \ldots, n+1)$ be an ordered set of real orthonormal functions defined over the interval $[a, b]$. The zeros of the $(n+1)$th orthonormal functions are x_r $(r = 1, 2, \ldots, n)$. Then an orthonormal set of n real n-dimensional orthonormal vectors can be constructed from the orthonormal functions by evaluating the lowest n orthonormal functions at the zeros of the $(n+1)$th orthonormal function. The orthonormal vectors are $\psi_r = [w_1\phi_r(x_1)\ w_2\phi_r(x_2)\ \cdots\ w_n\phi_r(x_n)]^T$ $(r = 1, 2, \ldots, n)$ in which w_r $(r = 1, 2, \ldots, n)$ are positive numbers.

This described orthogonal function conjecture is both unusual and a paradox. It is unusual because the zeros of the $(n+1)$th orthonormal function influence the construction of orthonormal vectors from the lowest n orthonormal functions. The orthogonal function conjecture is a paradox for reasons described in the next section.

Paradox

Let us now more closely examine the orthogonal function conjecture. The orthonormality conditions that the functions $\phi_r(x)$ satisfy are given by

$$\int_0^1 \phi_r(x)\phi_s(x)\,\mathrm{d}x = \delta_{rs} \qquad (r, s = 1, 2, \ldots, n+1) \tag{1}$$

where δ_{rs} is the Kronecker delta function ($\delta_{rs} = 0$ when $r \neq s$ and $\delta_{rr} = 1$) and x is defined over the interval $[0, 1]$. The zeros of $\phi_{n+1}(x)$ satisfy

$$\phi_{n+1}(x_t) = 0 \qquad (t = 1, 2, \ldots, n) \tag{2}$$

It is implied by Eq. (2) that $\phi_{n+1}(x)$ has n zeros. The n-dimensional orthonormal vectors are stated in the conjecture to satisfy the orthonormality conditions

$$\psi_r^T \psi_s = \delta_{rs} \qquad (r, s = 1, 2, \ldots, n) \tag{3}$$

where $\psi_r = [w_1\phi_r(x_1)\ w_2\phi_r(x_2)\ \cdots\ w_n\phi_r(x_n)]^T$ in which w_r shall be referred to as weighting constants. The paradox arises when we recognize that Eq. (3) represents a set of linear algebraic equations in terms of the unknowns w_r^2 $(r = 1, 2, \ldots, n)$. The number of equations is equal to n^2, and the equation corresponding to the pair of indices (r, s) is identical to the equation corresponding to the pair of indices (s, r). Therefore, the number of independent equations is $N = n(n+1)/2$. The paradox lies in that the number of equations

Received July 30, 1995; revision received May 10, 1996; accepted for publication Sept. 16, 1996.

*Professor, Department of Mechanical and Aerospace Engineering. Member AIAA.