JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 27, No. 4, July–August 2004

Simulated Annealing for Missile Optimization: Developing Method and Formulation Techniques

Ozan Tekinalp[*] and Muge Bingol[†]
Middle East Technical University, 06531 Ankara, Turkey

Hide-and-seek is a continuous simulated annealing algorithm that uses an adaptive cooling schedule. A number of improvements are proposed for the global optimum estimation required for the cooling schedule. To handle equality constraints, two approaches are examined: the rejection method and augmentation of constraints to cost using penalty coefficients. It is demonstrated that a faster convergence is possible if, in the penalty coefficients approach, equality constraints are replaced with tight inequality constraints. The missile trajectory optimization problem is formulated using nodes equally spaced in time until burnout and equally spaced in energy consumption after burnout. This approach is shown to be superior to the use of all nodes equally spaced in time. Also investigated is the effect of node number on the performance of the algorithm. The problem of combined optimization of design and control variables is also addressed. For this purpose a two-loop approach, where each loop has its own temperature and cooling schedule, is proposed, and its effectiveness is demonstrated.

Nomenclature

A_e, A_t, A_b	=	nozzle exhaust area, nozzle throat area, propellant burn area
AT	=	number of accepted trials during optimization
$\boldsymbol{b}$	=	vector of design parameters
C_L, C_D, C_F	=	lift, drag, and thrust coefficients
C^*	=	characteristic velocity of the propellant
D	=	aerodynamic drag force
f, f^*, $\hat{f}$	=	scalar cost function, and its global maximum, heuristic estimator
g	=	gravitational acceleration
h, h_f	=	altitude, specified final altitude
I_{sp}, I_t	=	specific impulse, total impulse
k	=	specific heat ratio
$k_1, \ldots, k_7$	=	penalty coefficients
L	=	aerodynamic lift force
m, m_t, m_p	=	instantaneous missile mass, total missile mass, propellant mass
n	=	pressure exponent used in burn rate calculations
P_a, P_e, P_{ref}, P_c	=	ambient pressure, exhaust pressure, reference pressure, chamber pressure
r, Range	=	horizontal distance from launch point, maximum range
$r_{P_{\mathrm{ref}}}$, r_{P_c}	=	propellant burn rates at reference pressure, chamber pressure
S_{ref}	=	reference area
T_D, T	=	thrust force at design altitude, thrust at current altitude
TU	=	number of temperature updates during optimization
t, t_b, t_f	=	time, burnout time, final time
t_{case}	=	case thickness
$\boldsymbol{u}$	=	vector of control variables
V, V_f	=	total velocity, impact velocity
$\boldsymbol{x}$	=	vector of state variables
α, $\boldsymbol{\alpha}$	=	angle of attack, vector containing angle of attack values of nodes
γ, γ_f	=	flight-path angle, flight-path angle at impact
η	=	impulse efficiency
θ	=	pitch attitude
ρ, ρ_p	=	density of air, density of propellant
σ_{steel}	=	yield strength of the steel
τ	=	temperature parameter used in simulated annealing algorithm
$\chi^2_{1-p}(d)$	=	$100(1-p)$ percentile point of the chi-square distribution with d degrees of freedom
$\boldsymbol{\omega}$, $\boldsymbol{\upsilon}$	=	vectors of optimization parameters

Introduction

TRAJECTORY optimization has been applied to aerospace vehicles ranging from rockets to aircraft.[1–4] Design optimization of aerospace vehicles integrates several disciplines, such as aerodynamics, structures, dynamics, controls, and propulsion.[5] Whereas many examples of multidisciplinary design optimization may be found in the literature, few address design optimization, including the trajectory of the vehicle, that is, controls and/or states.[6–9]

Various methods of the trajectory optimization of aerospace vehicles have been proposed. (A detailed review may be found in Ref. 10.) These methods are also used for multidisciplinary design optimization, where the following methods are identified: calculus-based methods, response surface methods, expert systems, genetic algorithms, simulated annealing, neural networks, and various combinations of them.[11] Calculus or gradient-based methods are well studied in the literature. They converge to a local minimum, however, and care must be taken to make sure that cost functions and constraints are smooth and differentiable. Nongradient methods have recently been applied to a number of problems, among which simulated annealing and genetic algorithms are most promising.[12]

Hide-and-seek is a continuous simulated annealing algorithm developed by Belisle et al.[13] It is capable of finding the global minimum of highly nonlinear functions. It is also said to be suitable for the optimization of nonconvex functions with disconnected regions, and it is guaranteed to converge to the global maximum with a probability of one (Refs. 13 and 14). The hide-and-seek simulated annealing algorithm was introduced to the aerospace community by Lu and Khan.[14] They used the algorithm to find optimum trajectories for a maneuvering aircraft. In their study, they also compared the algorithm to the principal axis method and Nelder–Mead simplex method and demonstrated its effectiveness. The algorithm is also applied to missile trajectory optimization and missile design optimization by

Received 24 April 2003; revision received 19 October 2003; accepted for publication 24 October 2003. Copyright © 2003 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0731-5090/04 $10.00 in correspondence with the CCC.

[*] Associate Professor, Aerospace Engineering Department. Senior Member of AIAA.

[†] M.S. Student, Aerospace Engineering Department; also Engineer, Engineering Development Department, Roketsan Missile Industries, Inc., Elmadag, 06780 Ankara, Turkey.

Tekinalp and Utalay,[9] and its effectiveness in solving highly nonlinear optimization problems was demonstrated. The purpose of this paper is to propose new approaches to the hide-and-seek optimization method and to the formulation of missile trajectory optimization and combined optimization of design and control variables.

The success of the hide-and-seek method depends on the accuracy of the estimation of global optimum. For this purpose, a heuristic estimator was proposed.[13] In this study, other approaches, such as use of an overestimate or use of the original heuristic estimator with bounds, are proposed. Their effect on the performance of the algorithm is investigated. hide-and-seek is an unconstrained optimization algorithm, and two approaches in handling nonlinear constraints are examined: the acceptance–rejection method and augmenting the constraints to the cost function using penalty coefficients. Also investigated is the effect of converting equality constraints into tight inequality constraints to be used for the constructing aggregate cost with penalty coefficients.

Trajectory optimization problems are converted into parameter optimization problems using control values specified at nodes. Because the number of nodes used has an important bearing on the trajectory optimization, the effect of the node number on the performance of the algorithm is investigated as well.

In previous missile trajectory optimization studies[6,7,9] the flight time after burnout was an optimization parameter. In this study, the termination of the flight is determined by the final total energy of the system calculated from the requirements on velocity and altitude at the terminal state. Consequently, the duration of unpowered flight is no longer an optimization parameter. Previously nodes were equally spaced in time. In the current study, the nodes are still equally spaced in time until burnout. However, after burnout, intervals between nodes indicate equal energy consumption intervals. The advantage of these formulations is also demonstrated.

Previously, flight control parameters and engine design parameters were optimized together. In this study, some parameters of the aerodynamic shape are also included. Because of the difficulties encountered, a two-loop (two-temperature) formulation of the optimization problem is proposed; results of the example problems are given and discussed.

In the following, first the combined design and trajectory optimization is discussed. Next, a brief survey on simulated annealing methods is given together with the hide-and-seek simulated annealing method. Some improvements are proposed, and implementation issues are discussed. Mathematical models used for trajectory optimization and design optimization are then given. The manuscript continues with the presentation and discussion of the study performed to investigate the effects of the improvements proposed to the optimization method and to the formulation of the missile optimization problems. Finally, conclusions are given.

Optimization Problems

Trajectory optimization problems are in fact optimum control problems. The problem may be formulated as follows:

$$\text{maximize } f(\boldsymbol{x}, \boldsymbol{u}, t) \tag{1}$$

$$\dot{\boldsymbol{x}} = \boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{u}, t), \qquad \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}, t) = \boldsymbol{0}, \qquad \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}, t) \leq \boldsymbol{0} \tag{2}$$

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0, \qquad \psi(\boldsymbol{x}_f, \boldsymbol{u}_f, t_f) = 0, \qquad \varphi(\boldsymbol{x}_f, \boldsymbol{u}_f, t_f) \leq 0 \tag{3}$$

Thus, the trajectory optimization problem is to find the control history $\boldsymbol{u}(t)$ such that a scalar performance index f is minimized [Eq. (1)] subject to differential equality constraints, as well as algebraic equality and inequality constraints on state and control variables [Eq. (2)], with specified initial conditions and constraints on terminal conditions [Eq. (3)]. Traditionally, trajectory optimization problems are solved using calculus of variations. Calculus of variations yields a two-point boundary-value problem.[15] This is called the indirect method.[10,16] However, the solutions of these two-point boundary-value problems usually require numerical techniques. Another method is to convert the optimal control problem into a parameter optimization problem.[17,18] This is called the direct method. For this purpose the value of the inputs $\boldsymbol{u}_i, i = 1, \ldots, l$, and/or states $\boldsymbol{x}_i, i = 1, \ldots, l$, at discrete points in time, called nodes, are sought. Two approaches are commonly employed: direct shooting and collocation.[10,16,17]

In multidisciplinary design optimization, the cost function and the constraints contain design variables as optimization parameters. Thus, the equations are already parameterized. However, with the vehicle trajectory included, the design optimization problem may similarly be posed as

$$\text{maximize } f(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, t), \qquad \dot{\boldsymbol{x}} = \boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, t)$$

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, t) = \boldsymbol{0}, \qquad \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, t) \leq \boldsymbol{0}, \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0$$

$$\psi(\boldsymbol{x}_f, \boldsymbol{u}_f, t_f) = 0, \qquad \varphi(\boldsymbol{x}_f, \boldsymbol{u}_f, t_f) \leq 0 \tag{4}$$

These equations may again be converted into a parameter optimization problem, as before, together with the design variables. Thus, the optimization parameters are both $\boldsymbol{u}_i, i = 1, \ldots, l$, and $b_j, j = 1, \ldots, k$.

Simulated Annealing Method

Simulated annealing is an optimization method that simulates the physical annealing process of finding the low-energy states of a solid at a particular temperature. For a given temperature τ, the probability of a system to be in state $\boldsymbol{r}$ may be found from the Boltzmann distribution $\exp[E(\boldsymbol{r})/k_B\tau]$, where $E(\boldsymbol{r})$ is the energy of the configuration and k_B is the Boltzmann constant (see Ref. 19). To simulate the annealing process, Metropolis criteria may be used. For this purpose, the change in the energy of a system with the movement of an atom is calculated, ΔE. If the movement lowers the energy of the system, it is accepted ($\Delta E \leq 0$). Otherwise, it is accepted with probability of $P(\Delta E) = \exp(-\Delta E/k_B\tau)$. In the simulated annealing optimization method, the cost function replaces the energy of the system, and the optimization parameters represent the atoms. This idea was first used by Kirkpatrick et al. to solve discrete combinatorial optimization problems.[19] The technique was later extended to the optimization of functions of continuous variables.[13,20–23]

In simulated annealing, the success of the algorithm to find the global optimum by the fewest number of function evaluations is closely related to the method used in selecting the next candidate point. For this purpose, various methods are proposed. For example, Vanderbilt and Lougie[20] propose selecting the next iteration point from a normal distribution and then multiplying it by the step size to find the next test value. Thus, a random walk with a fixed maximum step size is used. This maximum step size for each variable is updated after a predetermined number of trials. The new set of step sizes is selected proportional to the inverse of the Hessian calculated. Corona et al., on the other hand, displaces one variable at a time.[21] Siarry et al. also selects the next test point using random walk.[22] However, only randomly selected subsets of optimization parameters are displaced at each trial.[22] In Refs. 21 and 22, the step size is kept constant for a predetermined number of iterations. Others use pure random walk, where both search direction and step size are taken from uncorrelated uniform distributions.[13,23]

Another important aspect of simulated annealing is the cooling scheme used. Various schemes are also proposed in the literature. The most common approach is to cool by multiplying the current temperature by a fixed factor after a predetermined number of trials[20] or records.[21] Hajek[24] gives the necessary and sufficient conditions for a deterministic cooling schedule so that convergence to a global optimum is guaranteed. Siarry et al.[22] propose changing the temperature after a predetermined number of steps. Excess records lead to a large reduction in temperature. Otherwise, it is lowered by a small amount. Thus, two separate factors for fast cooling and slow cooling are used. Others propose an adaptive cooling schedule.[13,23] The cooling is carried out whenever a new record is found. The temperature selected depends on the distance of the record to the estimated global optimum. Thus, the temperature is small if the current record is close to the estimated global optimum. Otherwise, it is large.

Hide and Seek Algorithm[13]

Consider the following optimization problem:

$$\text{maximize}\, f(\boldsymbol{\omega}), \qquad \boldsymbol{\omega} \in S \tag{5}$$

where S is a compact body in $\boldsymbol{R}^d$. Thus, the objective is to find $\boldsymbol{\omega} \in S$ such that $f^* \equiv f(\boldsymbol{\omega}^*) \geq f(\boldsymbol{\omega})$ for all $\boldsymbol{\omega} \in S$.

First the Metropolis criterion shall be given:

$$\beta_\tau(\omega, \upsilon) = \text{minimum}(1, \exp\{[f(\upsilon) - f(\omega)]/\tau\}) \tag{6}$$

Then the algorithm proceeds as follows:

0) Choose a starting point $\boldsymbol{\omega}_0$ in the interior of S and a high enough starting temperature τ_0, and set $j = 0$.

1) Choose search direction $\boldsymbol{\theta}_j$ on the surface of a unit sphere with uniform distribution.

2) Choose step size λ_j from the uniform distribution such that $\Lambda_j = (\lambda_j \in R : \boldsymbol{\omega}_j + \lambda_j \boldsymbol{\theta}_j \in S)$. Set $\boldsymbol{\upsilon}_{j+1} = \boldsymbol{\omega}_j + \lambda_j \boldsymbol{\theta}_j$.

3) Choose V_j $(0 \leq V_j \leq 1)$ from a uniform distribution. Determine the next search point $\boldsymbol{\omega}_{j+1}$ from

$$\boldsymbol{\omega}_{j+1} = \begin{cases} \boldsymbol{\upsilon}_{j+1} & \text{if} \quad V_j \in [0, \beta_\tau(\boldsymbol{\omega}_j, \boldsymbol{\upsilon}_{j+1})] \\ \boldsymbol{\omega}_j & \text{if} \quad V_j \in [\beta_\tau(\boldsymbol{\omega}_j, \boldsymbol{\upsilon}_{j+1}), 1] \end{cases} \tag{7}$$

4) Update temperature, if $f(\boldsymbol{\omega}_{j+1})$ is greater than all previous function values. Otherwise, go to step 1.

The new temperature is calculated using

$$\tau = 2 \cdot \left[f^* - f(\boldsymbol{\omega}_j)\right] \big/ \chi^2_{1-p}(d) \tag{8}$$

This temperature update schedule gives an improvement in function value over the current iteration point with a probability of at least p. Because maximum value, f^*, cannot be known in advance, it is substituted with its estimate $\hat{f}$. The estimate is calculated using the following heuristic estimator:

$$\hat{f} = f_1 + \frac{f_1 - f_2}{(1-p)^{-d/2} - 1} \tag{9}$$

where $f_1$and f_2 are the two largest order statistics, respectively, and the parameter p corresponds to the probability that the real maximum f^* is larger than its estimate $\hat{f}$. The algorithm stops when the difference between the estimated value and the current value is small.

Application of the Method

The success of hide-and-seek that uses an adaptive cooling schedule depends on the accuracy of the estimation of global optimum. The heuristic estimator proposed in the original algorithm is based on current and previous records. If the current record is too far from the previous one, then the heuristic estimator gives unreasonably high values. Consequently, the temperature suddenly increases. The opposite may also happen and cause the temperature to drop prematurely. In either case, the algorithm may not converge to the actual maximum. However, use of an overestimate or a heuristic estimator with upper and/or lower bounds may also be reasonable and are examined in this paper.

Whereas simulated annealing algorithm is fundamentally an unconstrained optimization algorithm, it is also used for constrained optimization.[23] The usual approach is to discard the iteration point if the constraints are violated and to select another point in the same fashion. This approach is quite easy to apply if the constrains are simple upper and lower bounds on the optimization parameters. At a given iteration point, nonlinear constraints may also be evaluated, and the next trial point may be rejected if the constraints are violated.[23] However, the rejection method may be quite costly unless these constraints define a reasonably large region in the design space. The nonlinear equality constraints may also be handled by augmenting them to the cost function using penalty coefficients.[9,14] If the penalty coefficients are too large, the augmented cost may have very large peaks and wells, which makes the optimization difficult. On the other hand in many engineering problems, it is not essential to realize precisely the equality constraints, which may be replaced by tight inequality constraints. These approaches are investigated.

The effect of parameter number on the performance of the hide-and-seek algorithm was investigated previously by means of smooth multidimensional analytical functions.[13,14] Belisle et al.,[13] for example, with use of a highly nonlinear function with multiple local optima, have shown that, with true maximum used in Eq. (8), the number of records found, that is, temperature updates, increases linearly with function dimension. Lu and Khan[14] employed a similar function and showed that when true maximum is known, the function evaluation number also increases linearly. They also examined the effect of parameter number on the performance of the algorithm for trajectory optimization (a minimum-time, half-loop, fighter aircraft trajectory) reporting a nonlinear relation between the parameter number and the number of function evaluations. Additional results examining this relation are also given in the sequel.

Mathematical Models

Flight Mechanics Models

In this work a two-degrees-of-freedom missile model, assumed to always fly in trim flight conditions, is used (Fig. 1). Furthermore, an angle-of-attack autopilot realizes the commanded input instantaneously. Thus, pitch dynamics of the missile is not considered,

$$\dot{r} = V\cos\gamma, \qquad \dot{h} = V\sin\gamma$$

$$\dot{V} = (1/m)(T\cos\alpha - D) - g\sin\gamma$$

$$\dot{\gamma} = (1/mV)(T\sin\alpha + L) - g\cos\gamma/V, \qquad \theta = \gamma + \alpha \tag{10}$$

The engine is a solid propellant end burning rocket engine, and the total mass changes linearly with time,

$$m = m_t - (t/t_b)m_p \tag{11}$$

To calculate aerodynamic forces, necessary lift and drag coefficients of the hypothetical missile for different Mach numbers and angle-of-attack values are generated in a tabular form using Missile DATCOM software.[25] Because the missile is assumed to fly in trim flight condition, drag and lift coefficients change together with the center of mass position until burnout. Consequently, two separate tables, corresponding the fore and aft center of mass locations, are used. The coefficients corresponding to the current flight instant are found by interpolation. These coefficients also change after burnout, which requires application of separate table after burnout. Then, lift and drag forces are found from

$$D = \rho V^2 S_{\text{ref}} C_D \big/ 2, \qquad L = \rho V^2 S_{\text{ref}} C_L \big/ 2 \tag{12}$$

Whereas air density is a nonlinear function of altitude, the speed of sound is a function of the ambient temperature, which may be taken as a piecewise linear function of altitude.[26] Thus, during the integration of the equations of motion, each lift and drag coefficient table is also interpolated to find the coefficient values at the current angle of attack and Mach number. The motor is designed for specific exhaust pressure, and the change in thrust force with ambient pressure is taken into account,

$$T = T_D + (P_e - P_a)A_e \tag{13}$$

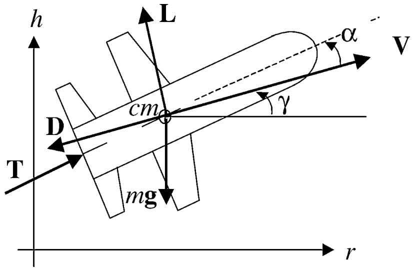


Fig. 1 Missile coordinate system.

The angle-of-attack values at the nodes are optimization parameters, which are linearly interpolated to find the corresponding angle-of-attack values between nodes during integration. The nodes are equally spaced in time. This does not constitute a major problem until burnout because burnout time is known before the simulation starts. However, the unknown total flight time is an optimization parameter. The terminal conditions specified by terminal velocity and terminal altitude give the final total energy (kinetic energy plus the potential energy) of the missile at impact. From the start of burnout, the missile continuously looses energy. For this reason, the equations of motion may also be integrated until the desired total energy achieved. Because the total energy is a parameter to be evaluated continuously, the nodes are placed with respect to total energy instead of time. Thus, the nodes after burnout indicate equal energy consumption intervals. The current angle of attack is again found by linearly interpolating from the neighboring nodes. It is also sufficient to impose the constraint on either terminal altitude or terminal velocity.

Design Models

The hypothetical missile used in this study has a cruciform wing and tail configuration and has an ogive nose. In design optimization, most of the missile geometric dimensions are taken constant. Wing semispan, wing root chord, wing taper ratio, nominal thrust, and burnout time are selected as optimization parameters. These parameters affect mass properties as well as aerodynamic properties of the missile. Nominal values of the optimization parameters as well as upper and lower bounds on them are given in Table 1. Numbers of designs are realized within the parameter bounds given. Tables of lift and drag coefficients needed for the trajectory simulations are obtained from Missile DATCOM[25] software. During design optimization, proper multidimensional interpolations are carried between these tables as well.

The engine is a solid propellant end burning rocket motor. The nozzle and engine geometry are shown in Fig. 2. The total propellant mass may be calculated in two ways. First, assume that the total propellant mass is consumed by burnout. Second, assume that combustion product is calorically perfect, expansion in the nozzle is isentropic, and maximum flow rate through the nozzle is achieved[27]:

$$m_p = A_b \rho_p r_{P_c} t_b = A_b \rho_p r_{P_{\text{ref}}} (P_c/P_{\text{ref}})^n t_b \tag{14}$$

$$m_p = I_t / I_{\text{sp}} g = T_D t_b \big/ \eta C^* C_F \tag{15}$$

Because Eqs. (14) and (15) must be equal, a relation between the thrust at design altitude and chamber pressure may be obtained. For engine design, nominal thrust T_D and total impulse I_t are selected as optimization parameters. Once they are specified, the chamber pressure as well as propellant mass may be calculated. For a given missile external diameter, burn area may be calculated by subtracting the case and liner thicknesses from this value. Liner thickness is constant (2 mm, in this example). However, case thickness, dependent on the chamber pressure, may be calculated using the hoop stress formula, which includes design pressure ratio (DPR):

$$t_{\text{case}} = \frac{P_c D_{\text{ref}} \text{DPR}}{2\sigma_{\text{steel}}} \tag{16}$$

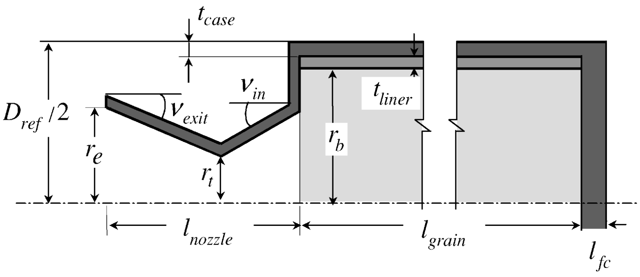


Fig. 2 Nozzle and engine geometry of the end burning solid propellant rocket engine: $A_t = \pi r_t^2$, $A_e = \pi r_e^2$, and $A_b = \pi r_b^2$.

The nozzle cross-sectional areas may be found from[27]

$$A_t = T_D / C_F P_c \tag{17}$$

$$A_e/A_t = [(k+1)/2]^{1/(k-1)} (P_e/P_c)^{1/k} \times \sqrt{(k+1)/(k-1)\left[1 - (P_e/P_c)^{(k-1)/k}\right]} \tag{18}$$

Nozzle inlet and exhaust angles are taken 45 and 15 deg, respectively. At this point all of the geometric dimensions of the engine and nozzle are known, and motor mass as well as its mass center may be calculated. Because wing shape is optimized, wing total mass and its center of mass are calculated using approximate formulas based on the wing planform area. Finally, total mass and mass centers of the missile before and after burnout are calculated.

Results and Discussion

In this section, results of the numerical experiments carried out to test the effectiveness of the improvements proposed for the hide-and-seek simulated annealing method as well as formulation techniques for missile optimization problems are presented. For this purpose three optimization problems considered. The first problem (P1) contains trigonometric functions and has only upper and lower bounds on parameters,[14]

$$\text{maximize}\left\{-\frac{\pi}{n}\left\{10\sin^2(\pi x_1) + \sum_{k=1}^{n-1}(x_k - 1)^2 \times \left[1 + 10\sin^2(\pi x_{k+1})\right] + (x_n - 1)^2\right\}\right\} \tag{19}$$

where $-10 \le x_i \le 10, i = 1, \ldots, n$. This function has 10^n local maximum and one global maximum of $f^* = 0$ at $x_i = 1$, $i = 1, \ldots, n$. The second problem (P2) is also an analytical function but with a nonlinear constraint besides simple upper and lower bounds[28]:

$$\text{maximize}\left\{(\sqrt{n})^n \prod_{i=1}^{n} x_i\right\}, \qquad \sum_{i=1}^{n} x_i^2 = 1$$
$$0 \le x_i \le 1, \qquad i = 1, \ldots, n \tag{20}$$

This function has a global solution of $f^* = 1$ at $x_i = 1/\sqrt{n}$, $i = 1, \ldots, n$.

The third problem (P3) is the maximum range trajectory optimization problem. A hypothetical air to surface missile (Table 1) is launched from a particular altitude with certain initial conditions ($V_0 = 237.3$ m/s, $\gamma_0 = 5$ deg, $h_0 = 8000$ m, and $\alpha_0 = 0$ deg) and certain impact conditions are required ($V_f = 271.2$ m/s, $\gamma_f = -75$ deg, $h_f = 1000$ m, $\alpha_f = 0$ deg). The nodal angle-of-attack values are constrained to the ±8-deg range. The equality constraints on the terminal conditions are also imposed, by augmenting the desired impact conditions to the cost using penalty coefficients, that is,

$$f = \text{range} - k_1 |V(t_f) - V_{\text{desired}}| - k_2 |\gamma(t_f) - \gamma_{\text{desired}}| \tag{21}$$

Table 1 Hypothetical missile parameters and upper and lower bounds used in the design optimization

Parameter	Nominal	Minimum	Maximum
D_{ref}, m	0.45	0.45	0.45
Length, m	2.57	——	——
m_t, kg	599	0	600
m_p, kg	166.6	——	——
Wing semispan, m	0.45	0.36	0.56
Wing root chord, m	0.55	0.52	0.81
Wing taper ratio	0.78	0.10	0.90
Tail gap, m	0.3	0.3	0.7
T_D, N	6000	2000	4500
I_t, kN · s	420	240	505
t_b, s	70	——	——

Table 2 Effect of estimator type and function dimension on the required number of function evaluations (problem P1) to attain $f > -0.1$

Estimator, f^*, type	$n = 2$	$n = 3$	$n = 4$	$n = 5$
Heuristic estimator, $\hat f$	14,786	2,030,888	4,954,106	>100,000,000
Overestimate, $f^* = 10$	36,381	352,760	8,758,305	22,528,464
Overestimate, $f^* = 1$	16,483	70,264	285,617	1,010,037
Heuristic with bounds, $-10 \le \hat f \le 10$	14,496	252,488	1,877,644	12,288,625
Heuristic with bounds, $-1 \le \hat f \le 1$	11,740	70,304	207,511	655,484
Global maximum	6,141	11,694	34,798	47,578

Table 3 Effect of the estimator performance of hide-and-seek algorithm for trajectory optimization

	Angle of attack sets					
Parameter	1	2	3	4	5	Average
Estimator 1, heuristic, $\hat f$						
FEN	18,661	7,165	15,827	17,990	30,000	17,929
AT	1,213	955	404	239	1701	902
TU	119	75	71	64	76	81
Range	64,663	60,864	69,201	66,236	39,433	60,079
Estimator 2, overestimate, $f^* = 75{,}000$						
FEN	30,000	30,000	30,000	30,000	30,000	30,000
AT	4,878	2,547	2,881	4,691	5,025	4,004
TU	71	71	87	96	84	82
Range	71,549	71,617	73,180	72,188	70,605	71,828
Estimator 3, underestimate, $f^* = 65{,}000$						
FEN	706	4,174	1,112	3,285	10,230	3,901
AT	127	371	145	362	1,056	412
TU	56	68	48	65	100	67
Range	65,656	65,801	65,042	65,430	65,089	65,404
Estimator 4, $\hat f \ge 70{,}000$						
FEN	30,000	30,000	30,000	18,032	30,000	27,606
AT	2,605	1,452	6,487	1,129	3,192	2,973
TU	129	91	89	78	136	105
Range	70,124	70,647	62,307	70,134	70,016	68,646
Estimator 5A, $10^5 \ge \hat f \ge 70{,}000$						
FEN	9,362	30,000	28,315	30,000	11,782	21,892
AT	673	836	939	1072	908	886
TU	88	100	130	111	83	102
Range	70,019	71,765	73,252	71,510	71,140	71,537
Estimator 5B, $10^5 \ge \hat f \ge 65{,}000$						
FEN	14,975	30,000	14,981	15,289	17,023	18,454
AT	355	778	891	467	494	597
TU	93	146	107	114	109	114
Range	66,530	72,619	67,403	69,779	71,571	69,580

In the baseline problem, k_1, and k_2 are taken as 500 and 1500, respectively. Initially these parameters were selected such that each entry in Eq. (21) is of comparable magnitude during optimization. After few runs, proper adjustments were made to the coefficients. Also note that terminal altitude may also be imposed instead of terminal velocity.

Estimation of the Global Maximum for Adaptive Cooling Schedule

As indicated before, the success of hide-and-seek depends on how well the global optimum is estimated. In this section, various estimators are examined through numerical experiments using P1 and missile trajectory optimization problems. Optimization of P1 of various dimensions ($n = 2, 3, 4, 5$) with three kinds of estimators are used. These are heuristic estimator proposed in the original algorithm [Eq. (9)], overestimate of the function value, and heuristic estimator with upper and lower bounds. The global maximum of the function is also employed. The overestimate uses a fixed value in cooling schedule [Eq. (8)], that is, $f^* = 1$ or $f^* = 10$. On the other hand, the bounded heuristic estimator is

$$f^* = \begin{cases} f_l, & \hat f \le f_l \\ \hat f, & f_l < \hat f < f_u \\ f_u, & \hat f \ge f_u \end{cases} \tag{22}$$

where, in this problem, (f_l, f_u) are taken as $(-10, 10)$ and $(-1, 1)$.

Because the maximum value of the function P1 is known, the optimization is carried out until the function value is greater than specified value of $f > -0.1$ (Table 2). The effect of estimator type on the required number of function evaluations to attain a particular closeness to the global maximum may be observed from Table 2, which reports the average results from 10 separate runs. It may be seen from Table 2 that runs with the true maximum require the fewest number of function evaluations. On the other hand, the heuristic estimator performed poorly and required a record number of function evaluations in all cases. Using an overestimate of the function value, instead of the heuristic estimator, usually performed much better than the heuristic estimator case. For function dimension 2 and 4, the heuristic estimator performed better than the overestimate ($f^* = 10$) because this value is not sufficiently close to the maximum (Table 2). In general, heuristic estimator with bounds, that is, $-10 < \hat f < 10$, performed better than the overestimate cases employing the upper bound of the former as the estimate, that is, $f^* = 10$.

The study is repeated for the maximum range trajectory optimization problem (P3) as well. The trajectory optimization programs were coded in Matrix-X[29] environment and were run on a Pentium III personal computer with 550-Hz clock speed. In this case, because the true maximum of this problem is not known, optimization is terminated when the temperature falls below a specified value. It is also terminated when the number of function evaluation (FEN) exceeds a predetermined value (in this study, 30,000). The optimization with 30,000 function evaluations required about 15 min on the cited computer. To evaluate proposed improvements properly each case is run five times, each time starting with a different initial parameter set. The sets contained 30 angle-of-attack parameters corresponding to 30 nodes. Of these 30 nodes, 10 of them correspond to the flight phase until burnout. The 11th node is at burnout. The 31st node is at impact, and its value is zero. For the results given in this section, terminal state constraints are very closely realized, final altitude is within ±0.5 m, final velocity within ±0.3 m/s, and final flight-path angle within ±0.2 deg.

First the heuristic estimator runs are conducted (estimator 1). The results of the optimization for each initial set are listed in Table 3. The FEN, accepted trials (AT), and temperature updates (TU), as well as maximum ranges achieved, are also given. These results show that in all cases except one the program was terminated automatically when the temperature dropped below the specified value. In the fifth set, the temperature did not drop and the program was terminated after completing all 30,000 function evaluations, achieving a poor 39.4-km range. The maximum range, 69.2 km, was achieved with set 3.

Although we do not know the actual maximum of a function, in many problems one may have an idea about the upper bound for the function value. The trial runs indicated that the maximum range of this missile was between 70 and 75 km. Consequently, the overestimate, 75,000, is selected as the function value (estimator 2). Results again given in Table 3 show that the achieved ranges are much better than those obtained with the pure heuristic estimator. On the other hand, because the estimate of the global maximum was high, the temperature did not drop to levels low enough to terminate the program. Consequently, in all cases, the program continued to loop until all of the permitted FEN was completed. Because the optimizations were terminated with function evaluation limit, the study is also repeated with an upper limit of 150,000 function evaluations. Again the five runs are conducted. The results ranged from 72.018 to 74.433 km, and the best range is attained after 144,305 evaluations. Average range was 72.924 km and average FEN when the maximums are attained was 94,112.

Because it is not always possible to guess a proper overestimate, the study is repeated with an underestimate value of 65,000 (estimator 3). The runs were terminated relatively early in all cases, with far fewer function evaluations (Table 3). The temperature dropped rapidly and attained a negative temperature at termination because, in all cases, the final range achieved was higher than the 65-km range specified. Ranges achieved with an underestimate were greater than those with a heuristic estimator and required far fewer function evaluations, although they were lower than those obtained using the overestimate. The heuristic estimator may initially estimate low values for the global optimum, which causes the program to spend excessive time at low function values before converging to the true maximum. By the prevention of the estimator from initially estimating low values, the temperature does not drop prematurely, which avoids program termination. The following estimator 4 is used:

$$f^* = \begin{cases} 70{,}000, & \hat{f} \le 70{,}000 \\ \hat{f}, & \hat{f} > 70{,}000 \end{cases} \tag{23}$$

The results given in Table 3 show significant improvement in convergence toward global optimum. In all cases except one, the range obtained was greater than 70 km. This is an excellent performance when compared with the heuristic estimator without a lower bound. (estimator 1). The temperature did not drop as fast, and in all cases except one the program terminated after completing the maximum FEN. Also note that the maximum range obtained is lower than that obtained by the overestimate.

As before, the heuristic estimator with upper and lower bounds is also tested. Whereas a lower bound prevents the premature drop in the temperature and termination of the program, imposing an upper limit on the heuristic estimator may minimize the poor performance of the program due to unexpected increase in temperature values. Then, for estimator 5A,

$$f^* = \begin{cases} 70{,}000, & \hat{f} < 70{,}000 \\ \hat{f}, & 70{,}000 < \hat{f} < 100{,}000 \\ 100{,}000, & 100{,}000 < \hat{f} \end{cases} \tag{24}$$

The results (Table 3) demonstrate further improvement in the best range is achieved (73,252 m) as compared with 70,647 m obtained with lower bound only. Similarly, average range also increased when using both bounds. The average number of function evaluations dropped from 27,606 to 21,892. Thus, the results show that although the upper bound is unrealistically high an upper bound on the heuristic estimator further improves the results substantially.

The lower bound of 70 km used in estimator 4 is also relatively high, and one may not be able to guess this lower bound properly. By the decrease of the lower bound value, its effect on performance may be examined. Thus, for estimator 5B,

$$f^* = \begin{cases} 65{,}000, & \hat{f} < 65{,}000 \\ \hat{f}, & 65{,}000 < \hat{f} < 100{,}000 \\ 100{,}000, & 100{,}000 < \hat{f} \end{cases} \tag{25}$$

The results (Table 3) indicate that reduction of the lower bound decreased the maximum range from 73,252 (obtained by estimator 5A) to 72,619 m. The average range also dropped from 71,537 to 69,580 m, and average function evaluation number dropped from 21,892 to 18,454. However, the results of estimator 5B are better than the results of the estimators 1, 3, and 4.

In summary it may be concluded that heuristic estimator with bounds performs better. The closer the bounds are to the true optimum, the faster the convergence of the algorithm will be. It is not detrimental if the optimum is not between these bounds. For example, if the optimum is below these bounds, we have an overestimate case, and program terminates after completing all permitted function evaluations. If it is above the bounds, the program will then terminate with a value better than the upper bound specified and with a negative temperature. Sometimes these bounds may be decided based on pure reasoning. For example, with optimized input parameters, a particular rocket will have a range longer than available, but probably will never reach a certain range, or a particular vehicle will be at least as heavy as those available on the market, but will never be lighter than a certain value, etc. An expert, or a person with sufficient experience will have better judgment. In fact, engineers solve the related problems many times during the development of the computer models, which helps them accumulate sufficient experience to make those judgments.

Implementation of Nonlinear Constraints

In the preceding missile trajectory optimization study, equality constraints were augmented to the cost using penalty coefficients [Eq. (21)]. In this section, the equality constraints issue is examined further through problem P2 [Eq. (20)] and a trajectory optimization problem. A usual approach is to augment the equality constraints to the cost through a penalty coefficient,

$$f = \left[\left(\sqrt{n}\right)^n \prod_{i=1}^{n} x_i \right] - k_3 \left(\sum_{i=1}^{n} x_i^2 - 1 \right)^2 \tag{26}$$

Another approach is to replace equality constraints with tight inequality constraints. In this study the following is used:

$$f = \left(\left(\sqrt{n}\right)^n \prod_{i=1}^{n} x_i \right) - k_3 \,\text{maximum} \left[\left(\sum_{i=1}^{n} x_i^2 - 1 \right)^2 - 0.01^2, 0 \right] \tag{27}$$

Whereas this form of constraint expression has no contribution to the aggregate cost when the result of the constraint is between ±0.01, it is continuous and penalizes the cost heavily as the difference exceeds these bounds. The penalty coefficient is initially selected as 2000, which is close to the ratio of the maximum value of the cost to the constraint when $n = 9$. The optimization is carried out for three, six, and nine dimensions using a heuristic estimator, a heuristic estimator with bounds, and the true maximum as the estimator. The optimization is stopped as soon as the function value attains $0.95 < f < 1.05$. Average results from 10 separate runs are presented in Table 4. With equality constraint formulation [Eq. (26)], resulting constraint costs were much less than ±0.01as expected. With inequality constraints, the results were again between the bounds given. Runs with equality constraints required many times more function evaluations to attain the desired accuracy than inequality constraints. The difference is especially pronounced for six and nine parameter cases. One may also confirm the findings of the earlier study, namely, the heuristic estimator with upper and lower bounds requires fewer function evaluations than heuristic estimator alone. Augmenting the absolute value of the equality constraint as done in trajectory optimization problem is also tested. However, because the actual cost

Table 4 Function evaluation numbers required for P2 to attain $0.95 < f < 1.05$[a]

	$n = 3$		$n = 6$		$n = 9$	
f^* type	Equality	Inequality	Equality	Inequality	Equality	Inequality
Heuristic estimator, $\hat{f}$	4,016	1,408	460,160	50,849	41,064,089	4,763,792
Heuristic with bounds, $0 \le \hat{f} \le 2$	732	636	366,888	39,388	20,943,375	4,067,623
Global maximum	728	543	36,611	15,094	314,287	79,450

[a] Average of 10 runs with equality and tight inequality constraints.

function is very high order, that is, nine, the problem converged very slowly.

The approach to replace equality constraints with tight inequality constraints is applied to missile trajectory optimization problem as well. For example, boundaries such as

$$\gamma_{\text{desired}} - 3 \leq \gamma(t_f) \leq \gamma_{\text{desired}} + 3\,\text{deg}$$

$$V_{\text{desired}} - 3 \leq V(t_f) \leq V_{\text{desired}} + 3\,\text{m/s} \tag{28}$$

are acceptable for the problem at hand. These inequality constraints are augmented to the cost with penalty coefficients,

$$f = \text{range} - k_1\gamma_{\text{cost}} - k_2 V_{\text{cost}} \tag{29}$$

where

$$\gamma_{\text{cost}} = \text{maximum}\left\{[\gamma(t_f) - \gamma_{\text{desired}}]^2 - 3^2, 0\right\}$$

$$V_{\text{cost}} = \text{maximum}\left\{[V(t_f) - V_{\text{desired}}]^2 - 3^2, 0\right\} \tag{30}$$

Note that the permitted variation in terminal velocity corresponds to an approximately ∓85 m change in altitude for the terminal conditions considered. Optimization is carried out using this aggregate cost function. In all cases the heuristic estimator with lower and upper bounds is employed (estimator 5A). The optimization is carried out five times as before. The new runs converged much faster than runs with equality constraints [Eq. (21)]. Thus, average number of function evaluations dropped, by about ninefold, from 21,892 to 2374. Although, an average lower maximum range is attained with tight inequality constraints (70.8 as compared to 71.5 km), terminal values obtained were within the desired range in all runs.

Another approach tested was to reject the trial point if the constraints are not satisfied. Because no trial point can be reasonably expected to satisfy equality constraints exactly, again a ±3 deg variation on flight-path angle and a ±3 m/s variation on the terminal velocity are permitted. The optimization program was run for each of the five initial angle-of-attack sets. However, after 5000 function evaluations, no trial point satisfied the constraints, and the program was terminated. Consequently, it may be stated that the acceptance–rejection method is not suitable for the current missile trajectory optimization.

It may be concluded that the performance of the algorithm improves substantially if the equality constraints are replaced by tight inequality constraints.

Effect of Parameter Number on Performance

The parameter number affects the performance of the hide-and-seek. This may easily be observed from the results of P1 and P2 presented in Tables 2 and 4. Increase in function evaluation number is especially formidable if same amount of closeness to the global maximum is required. Because the main parameter in trajectory optimization is the number of nodes used, its effect is investigated next.

Performance in Finding Maximum Range

First, the estimators proposed and given in the preceding section (Table 3) are examined with twofold increase in number of angle-of-attack parameters from 30 to 60. Whereas the original angle-of-attack values given in the sets are conserved, additional angle-of-attack values are inserted by taking the mathematical average of the neighboring nodes. Optimization, carried out for five different initial parameter sets, continues until a low enough temperature is achieved or maximum FEN exceeded a predetermined value. Equality constraints are employed [Eq. (21)]. Average results from these five runs are listed in Table 5.

When only the heuristic estimator is used, fewer function evaluations were needed with 60 angle-of-attack parameters than 30 parameters, although a shorter range was achieved. With an overestimate (estimator 2, $f^* = 75{,}000$) in all instances maximum allowable function evaluations were completed. Range average decreased from 71.8 with 30 parameters to 70 km with 60 parameters. Because more parameters require more function evaluations, a lower range is achieved with 60 parameter runs. In the underestimate case ($f^* = 65{,}000$), both runs achieved the required range (i.e. 65.4 km, and 65.2 km). However, 60-parameter runs completed 20,600 function evaluations on average, five times greater than required for 30-parameter runs (3901function evaluations). The 60-parameter runs with estimator 4 [Eq. (23)] again required more function evaluations, and a slightly higher range is achieved than with 30-parameter runs. Estimator 5A [Eq. (24)], 60-parameter runs terminated after completing all permitted function evaluations, and average range was 1500-m shorter than the 70-km lower bound. Similarly when estimator 5B [Eq. (25)] is employed, again about 50% more function evaluations are required, although a much lower range is attained in the average. In general, more nodes required more function evaluations.

Table 5 Average results from 30 and 60 angle of attack parameter runs using different estimators

Estimator type	Number of α parameters	FEN	AT	TU	Range	FEN at range
1	30	17,929	902	81	60,079	17,220
	60	14,947	1,061	148	55,456	10,521
2	30	30,000	4,004	82	71,828	23,996
	60	30,000	5,278	164	69,986	27,653
3	30	3,901	412	67	65,404	3,901
	60	20,600	2,058	181	65,186	20,600
4	30	27,606	2,973	105	68,646	25,408
	60	29,148	3,422	183	69,043	28,755
5A	30	21,892	886	102	71,537	21,725
	60	30,000	3,696	183	68,438	29,662
5B	30	18,454	597	114	69,580	18,266
	60	27,270	2,529	198	64,965	26,944

Table 6 Average results to show the effect of node number on FEN with different constraints

Number of α parameters	FEN for $f \geq 70{,}000$	f	FEN for range $\geq 70{,}000$	Range
		Equality constraint		
30	9,058	70,022	8,374	70,063
42	21,924	70,037	21,344	70,144
60	41,057	70,077	40,841	70,107
		Tight inequality constraint		
15	1,102	70,376		
30	1,610	70,128		
42	1,499	70,187		
60	918	70,113		

Performance in Attaining a Desired Range

Instead of waiting for the program to achieve global maximum, and not knowing whether it has achieved it, one may improve the basis of comparison by quantifying the program for the FEN required to achieve a desired value, close to maximum, as done for problems P1 and P2. Furthermore, other sets with different angle-of-attack values were included. First, sets with 15 angle-of-attack parameters were tested without conversion. Because using 120-element sets may require excessive function evaluations, it was decided to use 42-element sets instead. A heuristic estimator with double bounds was used [Eq. (24)] and equality constraints [Eq. (21)) were employed. Optimization is stopped as soon as the total cost and range exceeded 70,000 and corresponding FEN are recorded. On average, it required more function evaluations to achieve the desired cost value than to achieve the desired range because of cost-contained equality constraints on the terminal states (Table 6). From Table 6, it may be observed that, when parameter number is doubled, the FEN increased approximately fourfold.

Another numerical experiment was conducted to examine the effect of parameter number to achieve a desired range and cost, when tight inequality constrains are used [Eq. (29)]. The average results from 10 runs are listed in Table 6. The program was run until both the final range and final cost exceeded 70,000. Because the constraints were relaxed, it was easy to attain the desired value even with 15 nodes. Another interesting feature of the study is the drop in FEN when node number is increased from 30 to 60.

The overall evaluation of the preceding results indicates that, whereas node number is a parameter number that will increase the required FEN, more nodes also make it easier to approximate the optimum input history. In fact too few nodes prevented the program to attain the desired range when equality constraints are used. Thus, when node number is doubled, the required FEN does not increase as much as it increases with problems P1 and P2 (Tables 2 and 4). In fact, with relaxed constraints, because it becomes easier to attain a desired range with more nodes than with fewer nodes, required FEN become close to each other.

Comparison of Flight Time and Total Energy Formulations

In this section, flight time and total energy approaches are compared. For this purpose, a 30-parameter set (set 2) was run until the range and cost exceeded the target value of 70,000. An estimator with lower and upper bounds is used again (estimator 5A). Both equality and inequality constraints were examined [Eqs. (21) and (29)]. With equality constraints, the total energy approach required 6513 and 5705 function evaluations to exceed the target cost and target range, respectively. The flight time approach, on the other hand, required almost 15 times as many function evaluations than the total energy approach (90,073 and 89,914 function evaluations, respectively). When tight inequality constraints are used, the energy approach required only 515 function evaluations as opposed to 41,147 function evaluations required by flight time formulation. These results show that the total energy approach is superior to the flight time approach in trajectory optimization.

There may be two basic reasons for this. First, the range is very sensitive to time. Consequently, finding the precise time to realize the desired range is a difficult task, unless the time parameter is closely bounded to prevent it from taking a wide range of values during optimization. Unfortunately, without sufficient experience with a particular problem, one does not know how long the vehicle is going to fly beforehand. Second, with nodes describing equal energy consumption intervals, they are spaced more closely in time when there is large drag or lift force, which approximates the optimum input more accurately where it is needed most.

Missile Optimization Problems

Maximum Range Trajectory Optimization

When maximum range was taken as the baseline problem, various results have been presented. In this section, only the best trajectory obtained for the baseline missile is given and discussed. Identical launch and impact conditions are used. The maximum range at 1000-m altitude is found to be 73,252 m. Equality constraints on the terminal conditions are imposed [Eq. (21)], and the terminal requirements are very closely satisfied (Table 3, estimator 5A, set 3). The resulting trajectory is given in Fig. 3 together with trajectory of the same missile during ballistic flight.

To achieve maximum range, the missile initially climbs to over 18-km altitude before descending in a gliding trajectory. The missile flies with a positive angle of attack almost all of its flight (Fig. 4). Toward the final phases of flight, a high negative angle of attack (-8 deg) is employed to slow down the missile in the lateral direction and dive with a flight-path angle -75 deg is required. From Fig. 4, it may also be observed that, after burnout, time intervals between angle-of-attack values are not equal because nodes indicate equal energy consumption intervals.

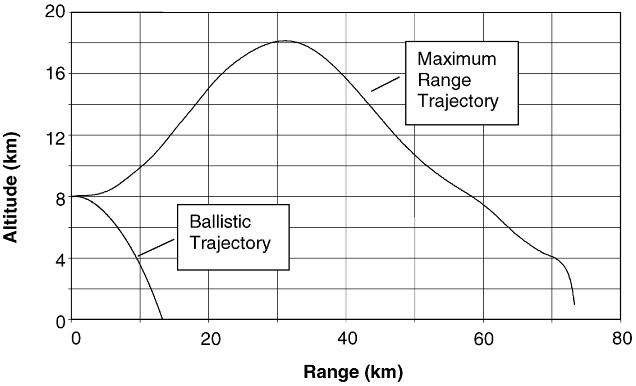


Fig. 3 Ballistic trajectory and maximum range trajectory.

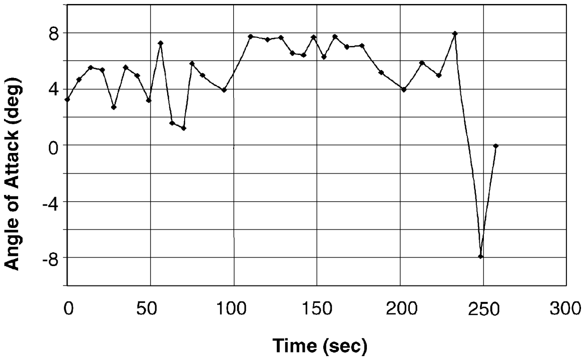


Fig. 4 Angle of attack history during maximum range trajectory.

The range of the same missile flying a ballistic trajectory is 13.3 km (at 39-m altitude), has a velocity over 1.16 Mach, and flight-path angle of -53 deg at impact. The flight finished after 43 s at which instant the engine was still alive.

Compared with the ballistic trajectory, a rather long range is achieved by optimizing the angle of attack history. However, how sensitive is the range to small changes in the input? For example, if there were angle-of-attack measurement bias, how would this bias affect the maximum range? To gain some insight to the problem, the nodal values of angles of attack are perturbed by a fixed amount. Flight simulation with $+0.5$ deg bias resulted in 67,862-m range, whereas -0.5 deg bias caused the missile to reach 68,146 m, that is, about 5.5-km shorter than the maximum range.

Combined Optimization of Control and Motor Parameters

This section addresses the problem of minimum weight design of a missile able to deliver a specified payload to a specified range. For this purpose, main engine parameters, nominal thrust T_D and total impulse I_t, are optimized together with the flight control parameters. When tight inequality constraints for the impact conditions are imposed, the augmented cost function becomes

$$f = -m_t - k_5\gamma_{\text{cost}} - k_6 V_{\text{cost}} - k_7 \text{range}_{\text{cost}} \tag{31}$$

where γ_{cost} and V_{cost} were defined earlier [Eq. (30)]. The range, on the other hand, is permitted to vary ±0.5 km, that is,

$$\text{range}_{\text{cost}} = \text{maximum}\left[(\text{range} - \text{range}_{\text{desired}})^2 - 500^2, 0\right] \tag{32}$$

Missile diameter, nose geometry, and mass of the warhead and guidance section remain unchanged. Wing geometry and location are identical. To speed up optimization, a database containing data for trim flight aerodynamic coefficients, and mass, for a given nominal thrust and total impulse was constructed. The database contained 35 sets of data generated from 35 designs, from seven total impulse and five thrust values. The selection of the range for trust and total impulse was based on the experience gained during the optimization runs. An estimator with lower and upper bounds ($-600 \leq f^* \leq 0$) is used, where the lower bound is the weight of the initial design. During optimization penalty coefficients k_5, k_6, and k_7, are taken as 30, 30, and 0.001, respectively. Optimization results are listed in Table 7, together with the properties of the original missile. The original missile, launched at 8-km altitude with 237.3-km/s speed, achieved a maximum range of 73,252 m. Engine parameter optimization reduced the total mass by more than 25% by dropping it from 599 to 439 kg. Though the total impulse also dropped slightly, the significant reduction was in the thrust value, which is dropped to more than half of its original value (from 6 to 2.7 kN). Because

Table 7 Combined optimization of control and motor parameters

Parameter	Original missile	$\text{Range}_{\text{desired}} = 73$ km Design 1	 Design 2	$\text{Range}_{\text{desired}} = 100$ km Design 3	 Design 4
V_0, m/s	237.3	237.3	170	237.3	170
m_t, kg	599	439	443	487	489
m_p, kg	167	203	203	253	254
I_t, kN · s	420	411	420	499	503
T_D, N	6,000	2,721	2,735	2,506	2,556
t_b, s	70	151	154	199	197
Range, km	73,252	73,207	72,952	99,540	99,587
$V(t_f)$, m/s	271.2	270.5	268.5	268.3	268.5
$\gamma(t_f)$, deg	−75.0	−73.4	−73.0	−77.9	−76.9
FEN	28,315	164	176	562	1,279

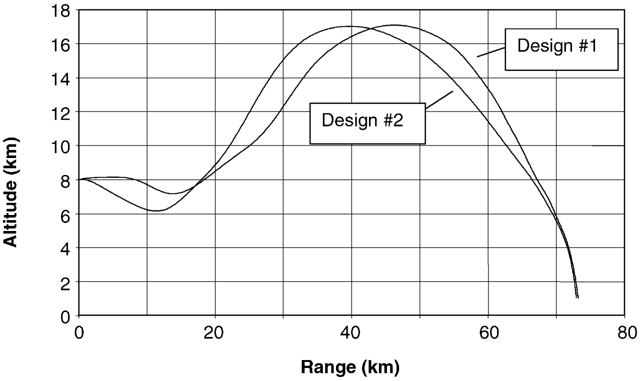


Fig. 5 Trajectories of the specified range minimum weight missile designs.

lower thrust is associated with lower chamber pressure, corresponding engine case mass and nozzle mass are also lower. On the other hand, the burnout time and total propellant mass are increased.

To examine the effect of the launch velocity, it is reduced to 170 m/s and the design process is repeated. The results (Table 7, design 2) show that the total mass has increased only by a mere 4 kg, mainly due to an increase in propellant mass. Burnout time also increased slightly. The resulting trajectories for these designs (Fig. 5) may be compared to the trajectory of the original missile (Fig. 3). Because these new designs have low nominal thrust, both missiles fall initially. Design 1 loses less altitude than design 2 because it has a higher initial velocity. The maximum altitudes reached are also lower than that of the original missile.

In another case, desired range was increased by 37% (100 km). Optimization results (Table 7) indicate that a 37% increase in range required about 10% increase in total mass, mainly due to the increase in the fuel mass. The nominal thrust also dropped, which causes a reduction in the weight of the engine case. However, total impulse and burnout times are increased about 20 and 30%, respectively.

The numbers of function evaluations needed during design optimization were lower for the lower range missile than the higher range missile. The 100-km-range missile launched at 237.3 m/s used almost all of the available impulse and required more function evaluations to converge. When a lower launch velocity was used (170 m/s), the program had convergence difficulties. It was overcome by a slight increase in the upper bound on total impulse, which results in the modification of the corresponding aerodynamic database. Optimization was repeated with this modified data. The missile with a higher desired range, launched with a lower velocity, used again almost all of the available total impulse (design 4). It also required more function evaluations (1279) because it was relatively difficult to find a proper trajectory with limited impulse.

Combined Optimization of Control, Motor, and Aerodynamic Shape Parameters

In this section, in addition to motor design parameters, the aerodynamic shape of the missile is also optimized to obtain a minimum weight missile that glides to its target on the most efficient flight path. To limit the number of parameters to be used in optimization, only the wing geometry is optimized because it has the most important bearing on the range of the missile. The selected wing optimization parameters are the wing root chord, semispan, taper ratio, and tail gap. The tail is located at the very aft of the missile. Consequently, its distance to the nose depends on missile length, itself dependent on engine design. To avoid meaningless designs, instead of wing leading-edge location, tail gap is selected as an optimization parameter. For the motor design, again thrust and total impulse, T_D and I_t, values are the design parameters. Again a database is constructed containing mass, engine, and aerodynamic properties of various designs. For the aerodynamic properties, Missile DATCOM[25] software is used as before. The database contained 21,875 sets of data, corresponding to seven total impulse values and five values for thrust, root chord, wingspan, taper ratio, and tail gap parameters. Each data set gave total mass and trim flight, lift, and drag coefficients. During optimization, linear interpolations are used to obtain necessary data from the database.

Optimization was carried out as before by use of using 30 angle of attack parameters, two engine parameters, and four parameters for wing geometry. The same launch conditions were specified, and the same impact conditions were sought. Although tight inequality constraints, as well as a heuristic estimator with upper and lower bounds, are used, the results were not successful. For example, the lightest missile to reach 73 km was found to have a mass of 485 kg. This is worse than the designs obtained by optimizing the engine parameters only. Similarly 100-km-range missile design yielded a mass of 473 kg, which is only slightly better than those given in Table 7 (design 3). Note that the latter case was repeated for 20 different initial parameter sets. In all cases, the optimization program terminated after a few hundred function evaluations, with very low temperatures. This suggests that there were many function improvements with very close values, and the temperature was dropping prematurely resulting in early termination of the program.

To force the program to a better solution, a two-loop (two-temperature) optimization approach was investigated. In the inner loop, the maximum range of a given missile configuration was found by optimizing the control parameters. The aggregate cost function for the inner loop is

$$f_{\text{inner}} = \text{range} - k_1 \gamma_{\text{cost}} - k_2 V_{\text{cost}} \tag{33}$$

In the outer loop, parameters related to the engine, T_D and I_t, and the wing geometry, semispan, root chord, taper ratio, and tail gap, were optimized to find the minimum-weight missile. The cost function of the outer loop is

$$f_{\text{outer}} = -m_t - k_7 \text{range}_{\text{cost}} \tag{34}$$

Each loop had its own temperature parameter with separate heuristic estimators with proper upper and lower bounds. Optimization results are given in Table 8. The total weight of the original missile is reduced from 599 to 401.7 kg (33% reduction, design 5), whereas its range is increased from 73.25 to 100 km (37%). When compared with an engine-only design optimization study, a 17.5% savings in weight may be observed for the missile that reaches a 100-km range. This increase in range is realized with an increase in wingspan and root chord, namely, wing planform area. Whereas the wingspan reached the highest value permitted (0.56 m), root chord was well below the permitted maximum because high aspect ratio wings are more efficient. The maximum range trajectory of the new design is given in Fig. 6.

The study indicated that the database is suitable for optimizing for a higher range as well. The optimization was repeated, this time for a 110-km range (design 6). A 10% increase in range was realized with a 7.5% increase in total mass, mainly due to the increase in propellant mass. The span, chord, and planform areas of the wings are quite close. Thrust, on the other hand, is increased slightly, with a 17.6% increase in total impulse.

Each multidisciplinary design optimization run took about 2 h on the personal computer mentioned earlier.

Table 8 Multidisciplinary design optimization results

		Range$_{\text{desired}} = 100$ km		Range$_{\text{desired}} = 110$ km
Parameter	Original missile	Design 3 control and motor parameters	Design 5 control, motor and wing parameters	Design 6 control, motor and wing parameters
V_0, m/s	237.3	237.3	237.3	237.3
m_t, kg	599	487	401.7	432.8
m_p, kg	167	253	169	196
I_t, kN · s	420	499	340.2	400.1
T_D, N	6000	2506	2686	2769
t_b	70	199	127	145
Range, km	73.25	99.54	100.2	109.7
$V(t_f)$, m/s	271.2	268.3	271.4	269.7
$\gamma(t_f)$, deg	−75	−77.9	−74.9	−73.9
Root chord	0.55	0.55	0.58	0.58
Taper ratio	0.78	0.78	0.70	0.70
Leading edge, m	2.29	2.29	1.81	1.92
Planform area, m^2	0.220	0.220	0.278	0.275

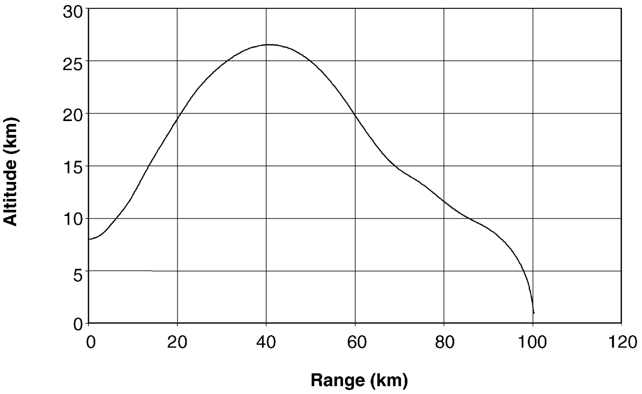


Fig. 6 Maximum range trajectory of design 5.

Conclusions

In this study, application of the hide-and-seek continuous simulated annealing algorithm to missile optimization problems is investigated. Improvements to the method and to the formulation techniques are proposed, and their effectiveness is demonstrated. It is shown that the hide-and-seek algorithm is very effective in solving such complex and nonlinear aerospace optimization problems. The strength of hide-and-seek lies in its adaptive cooling schedule. For this purpose, an estimate of the global optimum is needed. Various approaches are examined. Among them, the original heuristic estimator with lower and upper bounds is found to give good results.

Simulated annealing is an unconstrained optimization algorithm. Both the rejection method and augmentation of constraints to cost are tested. Also tested is the conversion of equality constraints to tight inequality constraints. It is found that the rejection method does not work for missile trajectory optimization. However, physically reasonable tight inequality constraints speed up convergence of the hide-and-seek algorithm. Furthermore, increase in node number in general increases the number of function evaluations for trajectory optimization.

Two formulations for maximum range trajectory optimization, total flight time and total energy approaches, are compared. It is demonstrated that the total energy approach, where the nodes after burnout are equally spaced in energy consumption, gives superior results. Hide-and-seek is also suitable for combined optimization of design and control variables. However, it may be necessary to carry out multiloop (multitemperature) optimizations to avoid a premature drop in temperature and termination of the process, due to the adaptive nature of the cooling schedule.

Acknowledgment

We would like to acknowledge Osman Merttopcuoglu, Roketsan Missile Industries, for his valuable comments.

References

[1] Hargraves, C., Johnson, F., Paris, S., and Rettie, I., "Numerical Computation of Optimal Atmospheric Trajectories," *Journal of Guidance and Control*, Vol. 4, No. 4, 1981, pp. 408–414.

[2] Jansch, C., Schenepper, K., and Well, K. H., "Ascent and Descent Trajectory Optimization of ARIANE V/HERMES," *Space Flight Mechanics*, CP-498, AGARD, Neuilly Sur Seine, France, 1990, pp. 3.1–3.27.

[3] Shaver, D. A., and Hull, D. C., "Advanced Launch System Trajectory Optimization Using Suboptimal Control," AIAA Paper 90-3413, Aug. 1990.

[4] Menon, P. K. A., Kim, E., and Cheng, V. H. L., "Optimal Trajectory Synthesis for Terrain Following Flight," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 4, 1991, pp. 807–813.

[5] Ashley, H., "On Making Things the Best—Aeronautical Uses of Optimization," *Journal of Aircraft*, Vol. 19, No. 1, 1982, pp. 5–28.

[6] Tekinalp, O., and Arslan, E. M., "Optimal Mid-Course Trajectories for Precision Guided Munitions with Collocation and Nonlinear Programming," *Proceedings of the 20th Congress of the International Council of Aeronautical Sciences*, Vol. 1, AIAA, Reston, VA, 1996, pp. 593–600.

[7] Tekinalp, O., and Arslan, E. M., "Multidisciplinary Design Optimization Tool for Air to Surface Missiles," *Proceedings of the ISSMO/UBCAD/UB/AIAA 3rd World Congress of Structural and Multidisciplinary Design Optimization* [CD-ROM], Paper 20MTA-3, University at Buffalo, School of Engineering and Applied Sciences, Buffalo, NY, 1999.

[8] Tsuchiya, T., and Suzuki, S., "A Study on Simultaneous Design/ Trajectory Optimization of Spaceplane," International Symposium on Optimization and Innovative Design, Japanese Society of Mechanical Engineers, Paper 158, July 1997.

[9] Tekinalp, O., and Utalay, S., "Simulated Annealing for Missile Trajectory Planning and Multidisciplinary Missile Design Optimization," AIAA Paper 2000-0684, Jan. 2000.

[10] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207.

[11] Frank, P. D., Booker, A. J., Caudell, T. P., and Healy, M. J., "A Comparison of Optimization and Search Methods for Multidisciplinary Design," AIAA Paper 92-4827, Sept. 1992.

[12] Hajela, P. "Nongradient Methods in Multidisciplinary Design Optimization—Status and Potential," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 255–265.

[13] Belisle, C. J. P., Romeijin, H. E., and Smith, R. L., "Hide-and-Seek a Simulated Annealing Algorithm for Global Optimization," Dept. of Industrial and Operations Engineering, Technical Rept. 90-25, Univ. of Michigan, Ann Arbor, MI, Sept. 1990.

[14] Lu, P., and Khan, M. A., "Nonsmooth Trajectory Optimization: An Approach Using Continuous Simulated Annealing," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 4, 1994, pp. 685–691.

[15] Byrson, A. E., and Ho, Y. C., *Applied Optimal Control, Optimization, Estimation and Control*, Balisdel, Waltham, MA, 1969, Chap. 2.

[16] Hargraves, C. R., and Paris, S. W., "Direct Trajectory Optimization Using Nonlinear Programming and Collocation," *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 4, 1987, pp. 339–342.

[17] Betts, J. T., and Huffman, W. P., "Application of Sparse Nonlinear Programming to Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 1, 1992, pp. 198–206.

[18] Hull, D. G., "Conversion of Optimal Control Problems into Parameter Optimization Problems," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, pp. 57–60.

[19] Kirkpatrick, S., Gelatt, C. D., and Vechi, M. P., "Optimization by Simulated Annealing," *Science*, Vol. 220, No. 4598, 1983, pp. 671–680.

[20] Vanderbilt, D., and Lougie, S. G., "A Monte Carlo Simulated Annealing Approach to Optimization over Continous Variables," *Journal of Computational Physics*, Vol. 56, No. 2, 1984, pp. 259–271.

[21] Corona, A., Marchesi, M., Martini, C., and Ridella, S., "Minimizing Multimodal Functions of Continuous Variables with the Simulated Annealing Algorithm," *ACM Transactions on Mathematical Software*, Vol. 13, No. 3, 1987, pp. 262–280.

[22] Siarry, P., Berthian, G., Durbin, F., and Hamesy, J., "Enhanced Simulated Annealing for Globally Minimizing Functions of Many Continuous Variables," *ACM Transactions on Mathematical Software*, Vol. 23, No. 2, 1997, pp. 209–228.

[23] Romeijn, H. E., and Smith, R. L., "Simulated Annealing for Constrained Global Optimization," *Journal of Global Optimization*, Vol. 5, No. 2, 1994, pp. 101–126.

[24] Hajek, B., "Cooling Schedules for Optimal Annealing," *Mathematics of Operations Research*, Vol. 13, No. 2, 1988, pp. 311–329.

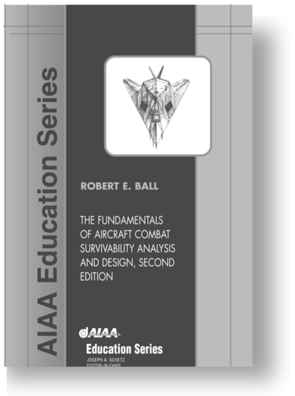
[25] Missile DATCOM Software, Revision 6/93, Wright Patterson AFB, OH, June 1993.

[26] Lan, C.-T. E., and Roskam, J., *Airplane Aerodynamics and Performance*, Roskam Aviation Engineering, Ottawa, KS, 1988, pp. 4–24.

[27] Sutton, G. P., and Bilarz, O., *Rocket Propulsion Elements—An Introduction to the Engineering of Rockets*, 7th ed., Wiley, New York, 2000, Chap. 3.

[28] Michalewicz, Z., and Shoenauer, M., "Evolutionary Algorithms for Constrained Parameter Optimization Problems," *Evolutionary Computation*, Vol. 4, No. 1, 1996, pp. 1–32.

[29] "Matrix-X User's Guide," Ver. 5, Integrated Systems, Santa Clara, CA, 1996.


AIAA Education Series
ROBERT E. BALL
THE FUNDAMENTALS OF AIRCRAFT COMBAT SURVIVABILITY ANALYSIS AND DESIGN, SECOND EDITION
AIAA
Education Series