JOURNAL OF SPACECRAFT AND ROCKETS
Vol. 37, No. 5, September–October 2000

Application of a Genetic Algorithm to the Optimization of Hybrid Rockets

P. L. Schoonover,* W. A. Crossley,† and S. D. Heister‡
Purdue University, West Lafayette, Indiana 47907

A genetic algorithm optimization technique has been successfully applied to the design of a large hybrid rocket booster. Optimizations to minimize gross liftoff weight or total inert weight have been carried out using a hybrid rocket sizing code developed at Purdue University. The genetic algorithm was able to find optimal or near-optimal designs that contained both continuous and discrete variables. Discrete variables included the propellant combination and the number of fuel ports, whereas the continuous variables, tank pressure, chamber pressure, and oxidizer massflux level, were simultaneously optimized using the genetic algorithm. Design solutions have been obtained from a discontinuous design space that contains a very broad, shallow minimum in weight. The resulting designs are discussed with some detail, illustrating their feasibility and some significant differences with previously published designs. The use of a genetic algorithm with a rocket sizing code appears to offer great potential to designers of rocket systems.

Nomenclature

c^* = characteristic velocity, ft/s
D = internal diameter, in.
G_{ox} = initial port oxidizer mass flux, lbm/(in.$^2 \cdot$ s)
I_{sp} = specific impulse, s
L_f = fuel grain length, in.
M_{inert} = total booster inert mass, lbm
M_{pl} = payload mass, lbm
N_{port} = number of ports in fuel grain
O/F = oxidizer/fuel ratio
P_c = chamber (or stagnation) pressure, psi
P_t = tank pressure, psi
r = fuel regression rate, in./s
Δv = velocity increment, ft/s
λ = propulsion system mass fraction

Introduction

DESIGN optimization of rocket propulsion systems remains a challenging, labor-intensive process. The complexity of the design space, involving numerous continuous variables such as system pressures, dimensions, and flow rates provides ample motivation to investigate advanced optimization tools. Compounding this problem is the desire (in many instances) to also optimize parameters that do not vary continuously over the design space. Examples of these discrete variables include the type of fuel or oxidizer, the number of nozzles or tanks, various candidate fabrication materials, or the number of fuel ports in a hybrid or solid booster.

Although there is little published information available on propulsion system design codes used in industry (for obvious competitive reasons), many of these codes are known to employ exhaustive searches or gradient-based schemes to optimize continuous variables. However, in most instances, optimal discrete variables can only be determined by a brute force approach in which systems are optimized assuming a fixed value of the discrete parameter (for example, a 15-fuel-port configuration), and the optimization is then repeated under a different assumption for the discrete parameter of interest. Obviously, this can be a labor-intensive process if more than a few values of discrete variables exist in the design space.

One promising approach that has recently been applied to this type of design problem is the genetic algorithm (GA).[1] The GA is a population-based search that relies on a Darwinist survival of the fittest strategy in which traits from the best performing designs in a given generation are passed on to designs in subsequent generations. Versions of the GA have been receiving wider use for engineering design problems, particularly in situations where the design space is discontinuous, multimodal, and involves a mixture of continuous, integer, and discrete variables.[2] Researchers have applied a GA to the conceptual design of rotary-wing aircraft,[3] where the configuration was included as a discrete variable (for example, a helicopter may incorporate a single main rotor or two main rotors in a tandem arrangement) with traditional continuous design parameters such as rotor loading and blade tip speed. This type of conceptual design problem is similar to the hybrid booster design problem.

To date, we are not aware of any published applications of the GA to the rocket design problem. For this reason, we were motivated to examine the performance of this tool in the design of a hybrid rocket booster. The hybrid booster provides an excellent example of a design problem involving continuous, integer, and discrete variables. In addition, over the past five years, we have developed a substantial design capability in this area through the creation of the hybrid rocket sizing (HYROCS) design code. These factors have provided the motivation for the efforts described in this paper in which a genetic algorithm is used to optimize hybrid booster designs in a discontinuous design space. The following section provides a brief description of the HYROCS design code, followed by a description of the GA approach to this problem and results for two design objectives.

HYROCS Code Description

Because substantial documentation on the HYROCS code has been published,[4–8] this paper provides only a top-level description. A flowchart for the algorithms and sizing modules used in the HYROCS code is depicted in Fig. 1. Separate modules are contained for sizing of the following components and systems: oxidizer tank, intertank structure, tank pressurization system (plumbing and fluid), turbopump, gas generator to power the turbopump, fuel tank for gas generator, injector, combustion chamber (including internal insulation and skirts), fuel section grain design, and nozzle. Figure 2 highlights a typical booster design and many of the components that are sized during this design process. The design methodology employed relies minimally on historical data; most component weights are calculated from the material volume necessary for adequate function of the part.

Presented as Paper 98-3349 at the AIAA/ASME/SAE/ASEE 34th Joint Propulsion Conference, Cleveland, OH, 13–15 July 1998; received 2 June 1999; revision received 26 April 2000; accepted for publication 26 April 2000.

*Graduate Student, School of Aeronautics and Astronautics. Member AIAA.

†Assistant Professor, School of Aeronautics and Astronautics. Senior Member AIAA.

‡Professor, School of Aeronautics and Astronautics. Associate Fellow AIAA.

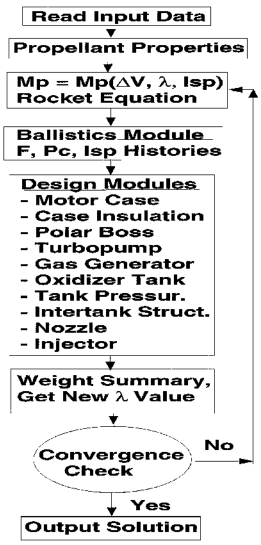

Fig. 1 Flowchart for HYROCS code.

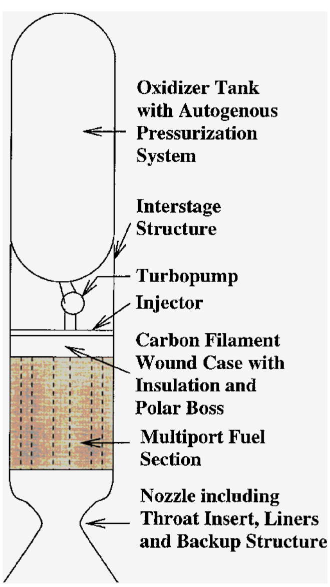

Fig. 2 Typical booster design denoting major components.

The nozzle design module is comprehensive; the nozzle is sized based on a simplified geometry comprised of four major components: a throat insert, exit cone liner, throat backup structure, and exit cone backup structure (or overwrap). The user has the option to select from several current state-of-the-art ablative materials in the throat insert and exit cone liner. These parts are sized based on predicted erosion (scaled from test data), charring, and strength conditions. Erosion rates are scaled from Titan booster data assuming convective heat fluxes vary as $P_c^{0.8}/D^{0.2}$. A one-dimensional flow calculation is performed to estimate the local pressure, Mach number, and temperature of the gases in the nozzle to support the heat flux calculations.

Various backup structural materials are available as well; these components are sized mainly to handle internal pressure and thrust loads on the nozzle. Insulators are designed under the assumption that they carry no structural load and overwrap/liner materials are sized to handle all structural loads imposed from the pressure distribution on the nozzle. The nozzle weight is determined from calculation of actual part volumes based on these sizing criteria. Specifics of the nozzle sizing are discussed in detail in Ref. 8.

An injector sizing module has been included as an upgrade to previously reported work. A thick-walled plate analysis is performed; high-strength steel is assumed as the fabrication material and material properties are degraded to account for the presence of injector orifices within the plate. Figure 2 highlights components sized in the routine.

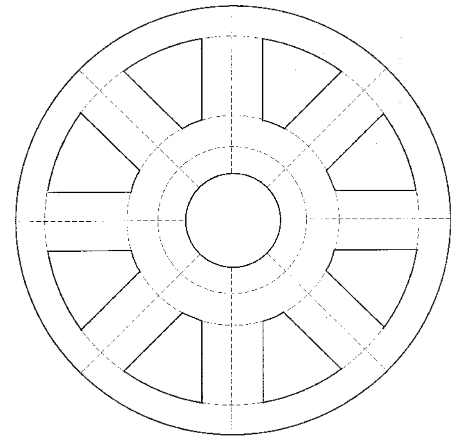

Fig. 3 Wagon-wheel fuel grain cross section.

The ballistics module, which is responsible for thrust and chamber pressure history calculations, as well as fuel grain sizing, is also quite comprehensive. Thermochemical data ($c*$ and vacuum I_{sp}) have been curve fitted as a function of mixture ratio O/F for the following propellant combinations: 1) liquid oxygen (LOX)/hydroxyl-terminated polybutadiene (HTPB), 2) 90% hydrogen peroxide (HP)/HTPB, and 3) nitrogen tetroxide (NTO)/HTPB.

Figure 3 shows the wagon-wheel fuel section geometry assumed in the sizing; port and burn surface areas are computed from actual port geometry at each instant in time. The same general arrangement, a center port and trapezoidal-shaped ports spaced at equal intervals about the circumference, is maintained for different numbers of ports. The minimum number of ports that can be supported with this grain design is four. Because of the lack of regression data on HP- and NTO-oxidized hybrids, the same regression rate correlation is assumed for all propellants:

$$r = 0.19 G_{ox}^{0.8} \Big/ L_f^{0.2} \tag{1}$$

An iteration is performed on the grain/case dimensions to accommodate the required mass of propellant and the burning time. Because the regression rate varies with time, the total web thickness consumed is initially unknown; this necessitates the iterative procedure.

The nozzle expansion ratio is set to ensure that the nozzle flow is not separated at the liftoff condition with the input chamber pressure level; the Kalt–Bendal[9] nozzle separation criteria provides an estimated separation pressure, and the nozzle exit pressure is set to be 3.0 psi above this limit. This condition sets the expansion ratio permitted for a given chamber pressure. Oxidizer flow is assumed to be constant over the firing duration.

Given this information, a complete time integration of the booster performance is conducted. Shifts in mixture ratio due to changes in oxidizer flow rate and port geometry are considered. The average vacuum I_{sp} value derived from this process is corrected to atmospheric conditions assuming an average altitude of 50,000 ft. during booster operation. At the present time, no trajectory integration is performed in the sizing process. Fuel and oxidizer residuals are considered as well as any oxidizer used as gas generator or tank pressurization fluid.

Hybrid Booster Sizing Process

The sizing process begins with the specification of a desired ideal Δv to be imparted to a given equivalent M_{pl}. It is important to provide a performance-based criteria for booster sizing; sizing to a specified total impulse will not necessarily meet mission constraints because the booster mass fraction also influences the velocity gain imparted to the vehicle. In the present study, a booster is designed to accelerate an equivalent payload of 219,000 lb to a velocity gain of 7728 ft/s. These values were selected to approximate the performance of the solid rocket boosters used on the Titan 34D launch vehicle. A minimum booster thrust/weight ratio of 2.0 and a 120-s burn time are also specified.

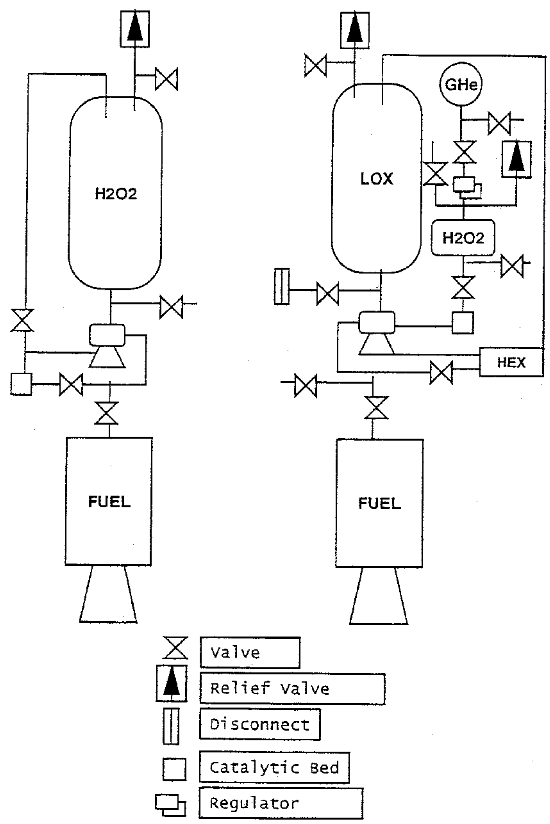

Fig. 4 Engine cycle comparisons; NTO-based cycle is analogous to LOX-based cycle shown.

Values for the design variables are set by the GA as described in the following section. Materials of construction are similar to those used in the Titan booster with the exception of a graphite–epoxy motor case for the combustion chamber. With this information, the sizing begins with assumed values for average I_{sp} and booster λ. The rocket equation provides the total amount of required propellant. The fuel and oxidizer masses are then determined based on the optimal mixture ratio required for the propellant combination of interest. Knowing the masses and volumes of both fuel and oxidizer permits sizing of the various inert components in the system. When the sizing is complete, an average I_{sp} and λ value can be computed, and the process can be repeated. Convergence of the sizing process occurs when the vehicle mass changes by less than 0.1% in successive iterations.

Figure 4 provides a schematic of the engine cycles for HP and LOX propellant combinations. The NTO/HTPB propellant combination incorporates an engine cycle identical to the LOX vehicle shown in the Fig. 4 schematic. The Fig. 4 schematic indicates the relative simplicity of the HP-oxidized cycle wherein the monopropellant characteristics of HP make it ideal for gas generator operation and tank pressurization. In contrast, the LOX-oxidized system requires a separate fluid for use in the gas generator. (Here HP was selected, but LOX/kerosene would be another obvious choice.) Because a separate tank is needed for the gas generator fluid, a separate pressurant tank is also required for the LOX- or NTO-oxidized systems. Finally, when LOX or NTO is utilized, a separate heat exchanger (labeled HEX in Fig. 4) is required to vaporize fluids for tank pressurization. These additional components have repercussions on inert weights as described in the results.

Genetic Algorithm Implementation

A GA works using analogies to the patterns of natural selection and reproduction that are displayed in biological populations. This concept originated in work presented by Holland for computer science applications,[10] was expanded by Goldberg[1] and others for engineering optimization, and is now becoming an accepted search and global optimization technique. Genetic algorithms, along with their underlying theory, are described in many references (for example, see Ref. 1), and so only a brief overview is given here.

One of the analogies to biological populations is the use of genes and chromosomes to represent each individual design in a population of designs. The variables and parameters that describe a design are mapped or coded into binary strings of ones and zeros; these strings are the genes of a design. This coding allows for simultaneous representations of discrete and continuous design variables in one optimization problem. In the present application, the optimization is performed using the discrete variables, propellant combination, and N_{port}, as well as the continuous variables G_{ox}, P_c, and oxidizer tank pressure P_t.

For this implementation, a binary coding was used to represent all design variables and parameters. Typical propellant combination and number of ports variable coding are shown in Tables 1 and 2, respectively. In a binary coding of the design parameters, the number of codes available is equal to 2^n, with n being the number of bits used to represent a parameter. This coding leads to a double coding of some parameter values in both Tables 1 and 2. There are three discrete propellant combinations under consideration here, but using two bits to represent propellant combinations provides four possible codes. Therefore, two different binary strings represent the HP/HTPB propellant combination. Similarly, 20 different integer values for the number of fuel ports are considered (6–25), which requires five bits to encode these values. With 32 possible strings for the number of fuel ports, several values for number of fuel ports are represented by two different strings. This double coding approach allows the GA to evaluate all possible binary strings. Obviously, for discrete or integer variables with 2^n possible choices, there is no need for double coding.

The continuous variables are also represented with binary strings, and so they are actually represented as discrete values. These variables are assigned a maximum and minimum value, and the resolution between each discrete value depends on the number of bits used to represent the variables. Table 3 summarizes the coding scheme

Table 1 Propellant combination variable coding

Binary string	Propellant type
00	LOX/HTPB
01	HP/HTPB
10	HP/HTPB
11	NTO/HTPB

Table 2 Number of ports variable coding

Binary string	Number of ports	Binary string	Number of ports
00000	6	10000	15
00001	6	10001	16
00010	7	10010	17
00011	7	10011	17
00100	8	10100	18
00101	9	10101	18
00110	9	10110	19
00111	10	10111	20
01000	10	11000	20
01001	11	11001	21
01010	12	11010	21
01011	12	11011	22
01100	13	11100	23
01101	13	11101	23
01110	14	11110	24
01111	15	11111	25

Table 3 Continuous variable coding

Variable	Minimum value	Maximum value	Resolution	Number of bits
P_c, psi	200	800	9.52381	6
G_{ox}, lbm/(in.$^2 \cdot$ s)	0.4	0.8	0.00315	7
P_t, psi	20	100	0.62992	7

used for the hybrid rocket continuous variables. Here a reasonably coarse resolution has been used; higher definition could be achieved by increasing the number of bits used.

Concatenating the strings for number of ports (5 bits), propellant type (2 bits), P_c (6 bits), G_{ox} (7 bits), and P_t (7 bits), forms a 27-bit chromosome that represents an individual hybrid rocket design. For example, a design with the chromosome 011001110100001101100001000 has 13 ports (01100), the NTO/HTPB propellant system (11), a chamber pressure of 580.9524 psi (101000), an oxidizer mass flux of 0.5701 lbm/(in.$^2 \cdot$ s) (0110110), and an oxidizer tank pressure of 25.0394 psi (0001000).

To begin a GA run, an initial generation of designs is created by randomly placing ones and zeros along the chromosomes for a given number of individuals. Deviating slightly from natural populations, this population size remains constant from one generation to the next. Maintaining a constant population size is not a strict requirement, but it makes for simpler coding of the GA. For the hybrid rocket design problem, a population size of about three times the total string length (80 individuals) was used based on guidelines in Ref. 11. Because of the double coding of the discrete variables as presented in Tables 1 and 2, the initial population will likely have a slight bias toward individuals with the doubly coded values. If these values are not desirable, the GA will eventually overcome this bias but will require more generations to do this.

The values of the design variables in each individual are decoded from the binary chromosome, and, from these values, a fitness value is computed for each individual. This fitness is analogous to the objective function value in a numerical optimization problem. Individual chromosomes with desirable fitness values are more likely to survive and become parents for a subsequent generation of designs. For the present studies, gross liftoff weight (GLOW) in pounds mass and total inert weight were used as fitness functions.

A tournament selection method is used to determine which of the individuals in a generation will survive to become parents of the next generation. In this selection approach, all individuals in the current generation are placed into a pot, and two individuals are randomly selected without replacement from this pot. The individual with the better fitness of this pair survives, and a copy of this individual's chromosome is placed into the parent pool. A second pair of designs is picked, evaluated, and the more fit individual is copied to the parent pool. This process of picking pairs from the pot and copying the best to the parent pool continues until the pot is empty, at which time the parent pool is only half full. The individuals in the current generation are replaced into the pot, and the tournament continues so that each individual competes twice. After this second round of the tournament, the number of strings in the parent pool will equal the population size. Figure 5 provides a flowchart describing this tournament selection process.

The tournament selection method possesses several advantages for the hybrid booster design problem. First, tournament selection allows for a lower fitness value to be considered the best, and so it directly provides minimization capabilities. Additionally, tournament selection compares two individuals at a time, rather than comparing the fitness of one individual against the entire population; this avoids problems with fitness scaling. Finally, tournament selection provides proper pressure toward better designs without allowing a few highly fit individuals to receive many copies in the parent pool. In the approach described, all individuals compete twice in the tournament. The best individual in the population always wins, and so it will have two copies in the parent pool. The worst individual will never win and will not appear as a parent.

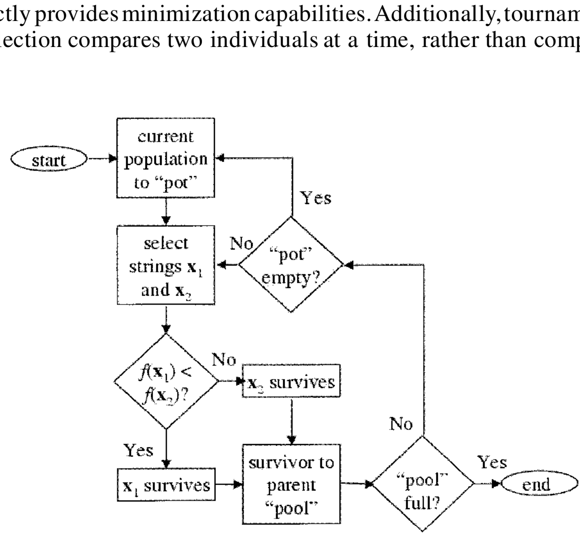

Fig. 5 GA tournament selection process.

Once parents have been selected, the next generation is formed via crossover and mutation processes. In the crossover process, two children are formed from two parents. The approach used in this effort is uniform crossover, in which each child string receives each bit from the first parent or the second with a 50% chance. For example, if a fair coin toss came up heads, the first child would inherit its first bit from the first parent and the second child would inherit its first bit from the second parent. If the toss were tails, the first child would inherit its first bit from the second parent, and the second child would inherit its first bit from the first parent. Each bit location is examined for crossover in this manner. Each resulting child chromosome is a combination of bits inherited from its parents. Because the parents were selected as good designs, combining chromosomes from the parents usually (but not always) result in better child designs. When a good child is formed, it will win its tournament, and then pass on parts of its chromosome to future generations. Poor child designs will not survive, and chromosome patterns corresponding to poor performance will be removed from the population. Through this process of combining good chromosomes to form children and then selecting the best of these children as parents for the next generation, the GA evolves patterns of ones and zeros in the chromosomes that correspond to designs with good fitness values. This process provides most of the GA's optimizationlike behavior.

The crossover operator does much of the work to find optimal designs, but the mutation operator assists in the design space search. After the child strings have been formed, a mutation may occur that will change a bit to its binary opposite, for example, a zero becomes a one, or vice versa. This mutation can allow for a new binary pattern to be introduced in a chromosome that was not present in a child's parents. If this mutation results in a high-fitness design, then this trait can be passed on to future generations. Conversely, if the mutation results in a poorly performing individual, that individual is unlikely to survive, and the binary pattern created by the mutation is removed from the population. As in nature, the mutation process in the GA occurs with very low frequency. The probability of mutation was 0.00875 for the hybrid rocket problem.

To assign fitness values, the child design chromosomes are decoded and evaluated for GLOW or M_{inert} using the HYROCS code. The selection operator is used to pick individuals from this group to become parents for the next generation, and the process continues through crossover and mutation to form new generations. With the selection operator favoring good individuals and the crossover operator combining features of good individuals, the GA population moves toward the globally optimal design. For this work, the genetic algorithm is halted after 100 generations.

Because no derivatives are used and the search is conducted with a population of design points rather than a single point-to-point search, the GA is unlikely to stop its search at local optima. However, with discretized variables, probabilistic operators, and no gradient information, the GA is not able to guarantee that it has found the exact global minimum. The best-fitness individual encountered during the run of the genetic algorithm then provides a near globally optimal design. For computational simplicity in this implementation, a pseudorandom number generator is used for the probabilistic operations. For a given seed number, this generates a list of random numbers; for the same seed, the same list will be generated. To address how the GA's stochastic processes affect the results, several runs were conducted with different seed numbers for each fitness function of interest.

Results

As mentioned earlier, both total inert mass M_{inert} and GLOW optimizations have been conducted using the GA methodology described in the preceding section. A total of 100 generations were evaluated using a population size of 80 individuals. For the M_{inert} runs, the number of fuel ports ranged from 6 to 25 as described in Table 2, and so a 27-bit chromosome was used. The GLOW runs encoded the number of fuel ports between 10 and 25, resulting in a 26-bit chromosome, thereby eliminating double coding of the N_{ports} variable. All other variable coding and GA parameters were the same for both cases.

Typical run times were on the order of 10–15 min on a 200-MHz Pentium personal computer. The GA required 8000 fitness evaluations (80 individuals times 100 generations) to complete each run, which appears large compared to traditional optimization techniques. As a comparison, however, a complete enumeration of the design space available to the genetic algorithm would require over 134,000,000 function evaluations (2^{27}) for the problem to minimize M_{inert} using 27-bit chromosomes. Similarly, the problem to minimize GLOW would require over 67,000,000 (2^{26}) function evaluations.

Because the problem statement addressed here combines discrete, integer, and continuous variables, an exact comparison with gradient-based search techniques is not possible. However, an approximate comparison is possible. The M_{inert} problem incorporates three propellant systems and 20 different numbers of fuel ports; this means there are 60 possible combinations of discrete and integer variables. A traditional gradient-based search could be conducted for each of these 60 combinations. If the traditional technique requires 135 or more function evaluations to determine function values and numerical derivatives for the search, the computational cost (60 combinations times 135 function evaluations equals 8100 evaluations) will exceed that of the genetic algorithm approach. Additionally, gradient-based optimization routines halt at the local minima nearest the initial design, so conducting several optimizations using different initial designs is common practice.

Figure 6 demonstrates the performance of the GA in finding an optimal number of fuel ports, one of the two discrete variables, during a run to minimize M_{inert}. There are 80 individuals in the population, and the plots in Fig. 6 show the number of fuel ports associated with each of these designs. The individual designs of the randomly generated population (generation 1) have scattered values for the number of ports as expected. By the 15th generation, the GA has limited its search to the upper $\frac{2}{3}$ of the optimization range because designs with a small number of fuel ports have larger M_{inert} values and are not selected to survive as parents of future generations. By the 100th generation, 74 of the 80 individuals have 16 fuel ports. Four of the six outliers in generation 100 have either 15 or 17 ports, very near the optimum value.

Figure 7 depicts the search of the GA in a plane of the design space containing two of the continuous variables, G_{ox} and P_c. These plots show all 80 designs in the population at selected generations during the run of the GA; each design is represented by a +. The randomly generated initial population (generation 1) shows designs scattered throughout the possible design space. Booster designs with high P_c are associated with large M_{inert} values, so that the population evolves to the left half of the design space after only five generations. By the 100th generation, nearly all individuals share the optimal values of P_c. Figures 6 and 7 illustrate that the GA approach successfully finds values for continuous, discrete, and integer variables.

The convergence-like behavior of the GA for a M_{inert} run is illustrated in Fig. 8. In the Fig. 8 plot, the fitness value of the best design (lowest M_{inert}) in each generation is shown. As the generations progress, the lowest M_{inert} value decreases rapidly over the first several generations and reaches a nearly convergent state by the 40th generation. The design space appears to exhibit a very broad and shallow region near the optimum because M_{inert} of the best individual from a given generation varies only about 4% over the 100 generations evaluated by the GA. The GLOW optimization runs exhibited similar trends with a rapid improvement in the fitness function over the first 10–20 generations and with slower fine tuning over the remaining generations.

Because mathematical convergence to an optimum cannot be guaranteed with the GA, three runs with different initial seeds were conducted for each of the two objective functions. The best designs encountered during these runs are summarized in Table 4. In the GLOW optimizations, two of the three seeds converged to the same optimum, whereas the third case (seed 2) came up with a slightly different optimum. The inert mass optimizations show more variation in the design parameters. In particular, seed 2 found an optimum

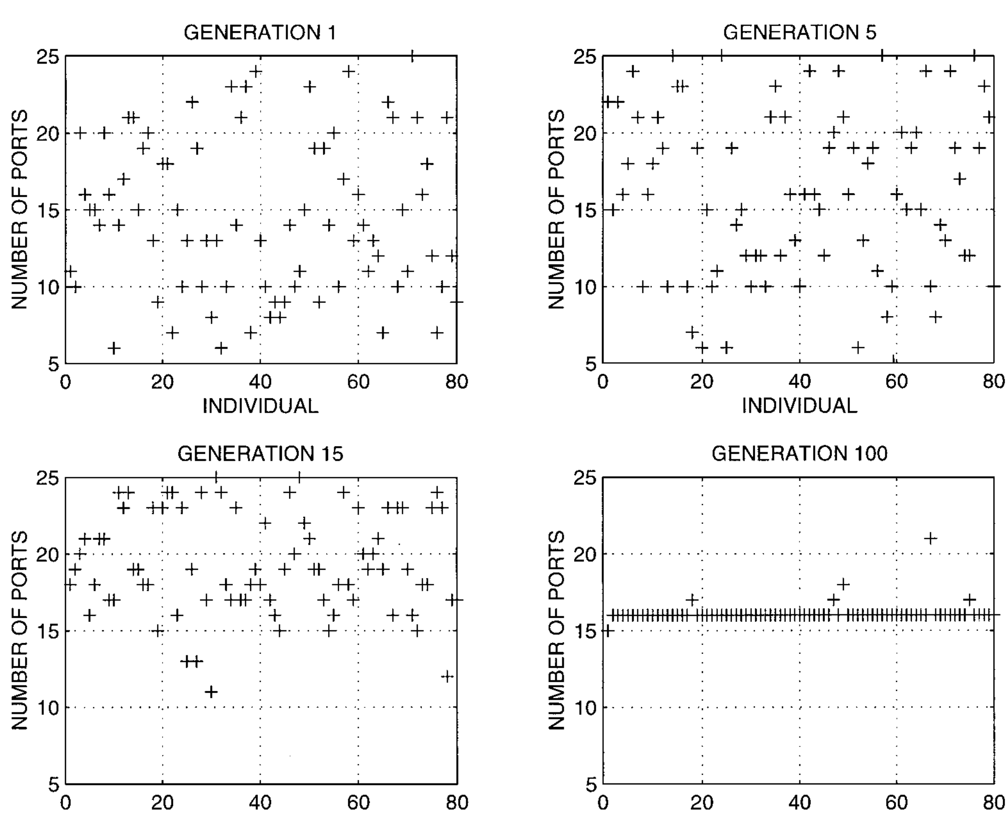

Fig. 6 Convergence of the genetic algorithm to optimal number of fuel ports.

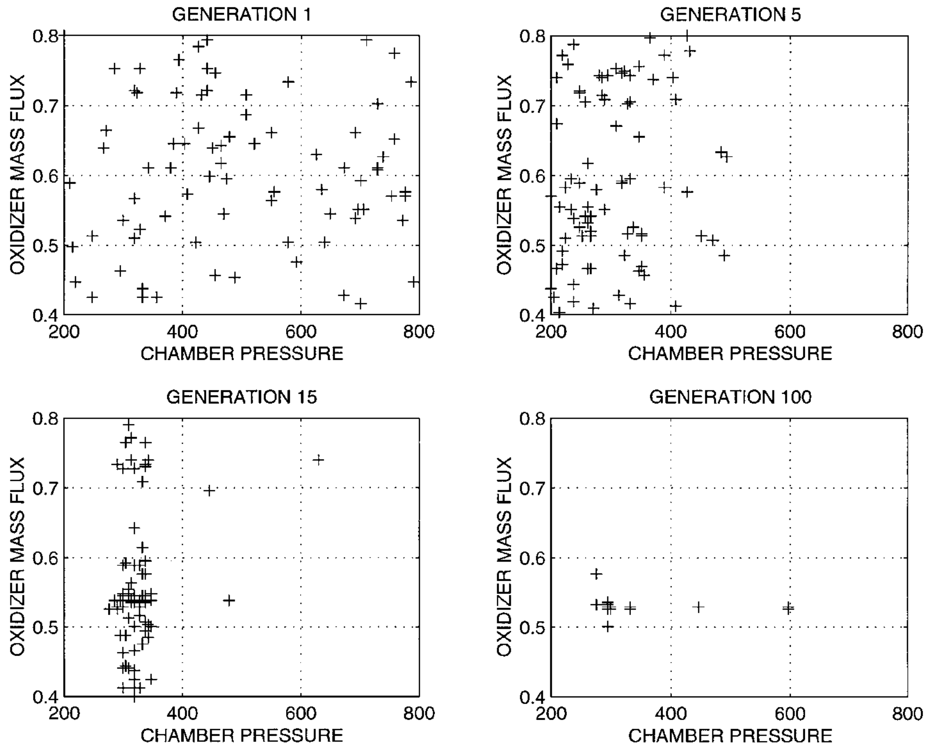

Fig. 7 Convergence of the genetic algorithm to optimal P_c (psi) and G_{ox} [lbm/(in.^{2} ·s)] values.

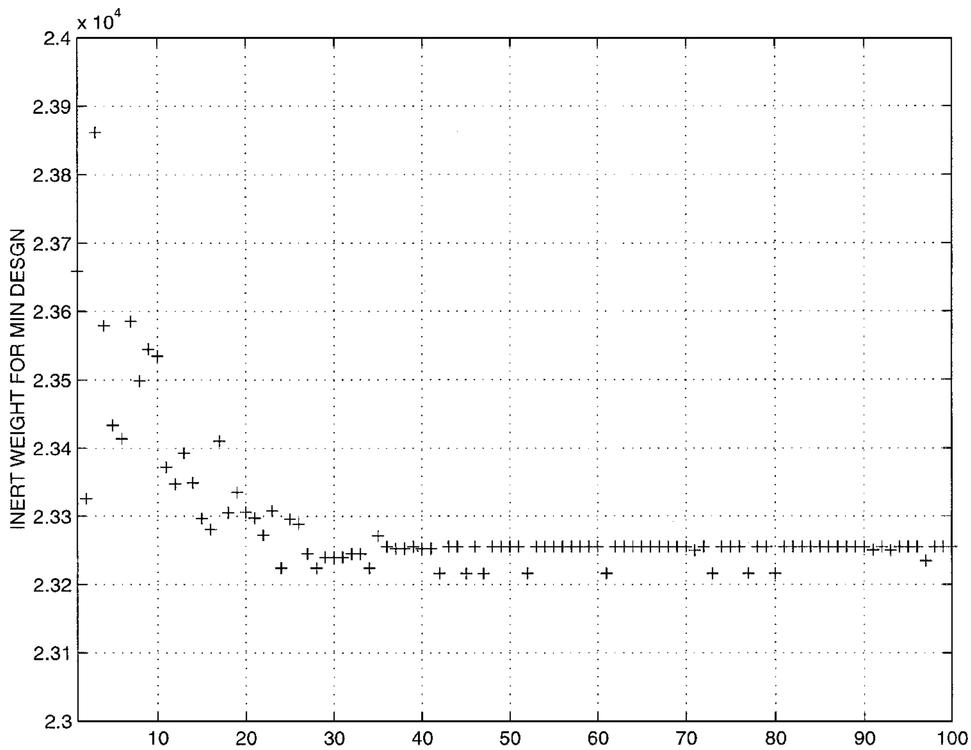

Fig. 8 Convergence history of one M_{inert} optimization run.

with considerably fewer ports than in the other cases. However, all of the optima differ in inert mass by less than 0.1%; this is the overall mass convergence tolerance employed in the HYROCS code. Because all optimum values lie within the accuracy of the sizing code itself, the GA has shown excellent performance for these problems.

The results of Table 4 also illustrate that the GA is capable of exploring different parts of the design space. The difference in GLOW is very small for designs with 15, 10, and 16 ports, which implies that these designs have nearly the same performance. In practice, the manufacturing costs and complexity of a hybrid rocket increase with the number of ports, and so the 10-port design would be a favorable choice to address manufacturing concerns. Generating different, but similar performing, designs with the GA allows a designer to take into account characteristics that were not initially considered in the problem statement.

For the GLOW optimization, the seed 1 result was obtained in just the 34th generation. Although it is uncommon to encounter the best design so early in the optimization process, this is not abnormal.

Table 4 Performance of the GA for different initial seeds

Seed	No. port	P_c, psi	G_{ox} lbm/(in.$^2\cdot$s)	P_t, psi	M_{inert}, lbm
			M_{inert} optimizations[a]		
1	15	295	0.526	55.3	23,255
2	10	295	0.403	60.9	23,275
3	16	295	0.526	50.9	23,253
			GLOW optimizations[b]		
1	11	205	0.409	59	54,0929
2	10	205	0.441	66	54,0493
3	11	205	0.409	59	54,0929

[a]All M_{inert} seeds found HP/HTPB propellants to be optimal.
[b]All GLOW seeds found LOX/HTPB propellants to be optimal.

Because the GA uses a random initial population and probabilistic operators, it is certainly possible to find the best design early in the search procedure. In the case of the seed 1 run, this best individual certainly won its tournament competition, but because this design was found early in the search, this design's mate was certainly lower performing in comparison. The exact genetic pattern to this individual's excellent fitness was lost during the mating process and not encountered again, but the designs present in the final generation had fitness values nearly equivalent to the best-ever value.

Physical Significance of Results

Table 4 shows that both optimizations arrived at P_c values in the 200–300-psi range. It is not surprising that similar chamber pressure values were selected even though the combustion chambers are of differing dimensions with the different propellants because both cases used the same construction materials. In earlier work,[5] which used exhaustive search techniques to find optimal values, substantially higher values of P_c (in the 1000-psi range) resulted. However, these previous efforts did not include the effect of injector mass, a substantial component that grows rapidly in weight with increased chamber pressure; injector mass was included in the present work. In addition, improvements to the motorcase design modules have tended toward larger case weights than those predicted in previous studies. Therefore, the combustion chamber becomes a more significant fraction of the total inert weight, thereby driving the optimum to lower P_c values.

The tank pressures tended to gravitate to the 55–65-psi range, which is consistent with existing propulsion systems. The optimal G_{ox} values obtained are in the range of those used by others conducting hybrid rocket design and experimentation, and these values are consistent with those obtained in earlier studies.[5] Here, the tradeoff in optimizing this variable is subtle. Large G_{ox} values produce large regression rates [Eq. (1)] and lead to substantial changes in fuel port geometry over the course of the firing. Because the fuel surface area is changing with this port geometry, fuel flow rates can shift substantially during the burn, thereby leading to shift from the optimal O/F ratio. If one uses a low G_{ox} value, very large fuel ports (and, hence, combustion chambers) are required, thereby driving up inert weight. The optimal values obtained in this study are similar to those obtained in a previous study[7] using exhaustive search techniques.

Tables 5–7 provide performance, size, and weight comparisons for optimal vehicles designed under minimum GLOW and minimum M_{inert} objectives. In Table 5, the I_{sp} values are computed assuming an atmospheric pressure equivalent to 50,000-ft altitude. The superior I_{sp} value of the LOX oxidizer system results in a lower GLOW than other propellant sytems. The HP-oxidized vehicle has an I_{sp} about 10% lower than the LOX-oxidized design, a factor leading to the requirement for higher propellant flow rates in the HP-based vehicle. The high mixture ratio required for the HP/HTPB combination leads to increases in gas generator flows as well because more fluid must be pumped for this propellant combination.

The higher density of the HP/HTPB propellant combination leads to a reduction in overall vehicle length as indicated in Table 6. Differences here arise because HP has about a 25% higher density than LOX and because the LOX-based chamber needs to be very long to accommodate the additional fuel required for the lower mixture ratio operation of this propellant combination. The nozzle exit diameter for the LOX-based vehicle is significantly larger than the booster because the booster diameter is set by internal ballistics constraints and the grain design. This situation would lead to high drag, a factor not presently considered in the optimization. The M_{inert} optimized vehicle has a larger diameter to accommodate the higher flow rates and larger number of fuel ports.

Table 5 Performance for GLOW and M_{inert} optimized boosters

Item	GLOW	M_{inert}
Oxidizer	LOX	HP
Chamber pressure, psi	205	295
Ullage pressure, psi	60	55
Oxygen flowrate, lb/s	1708	2475
Gas generator flowrate, lb/s	5.3	9.7
Delivered vehicle I_{sp}, s	310	278
Expansion ratio	4.44	6.15
Mixture ratio	2.5	7.5

Table 6 Dimensional comparison

Item	Dimension, in. Glow	M_{inert}
Overall length	1218	1011
Overall diameter	94	116
Chamber length	526	254
Chamber thickness	0.134	0.242
Injector thickness	0.65	0.97
Tank length	531	572
Tank thickness (cylinder)	0.33	0.39
Nozzle throat diameter	55.7	46.1
Nozzle exit diameter	117	114

Table 7 Weight comparison

Component	Weight, lbm Glow	M_{inert}
Usable propellant	291,514	340,968
Residual propellant[a]	5,612	6,406
Total propellant	297,126	347,374
Oxygen tank pressurant	1,041	750
Oxidizer tank	7,975	7,855
Intertank	515	865
Combustion chamber	6,636	5,206
Chamber polar bosses	146	168
Nozzle	5,811	5,438
Turbopump	415	1,041
Gas generator system[b]	1,828	1,165
Injector	275	749
Misc. structure	1,196	807
Total inert weight[c]	24,797	23,295
GLOW	540,929	589,668
Booster mass fraction	0.923	0.936

[a]Includes oxidizer tank pressurant.
[b]Includes gas generator and its operating fluids and a heat exchanger for LOX-based systems.
[c]Excludes residual propellants.

The weight summary indicates that the HP-based design's simplicity explains the tendency of the GA to choose this oxidizer when minimizing inert weight. The heat exchanger is not needed in the HP design because the tank is pressurized with decomposed peroxide from the gas generator system. In the LOX-based vehicle, a separate fluid must be carried for the gas generator. The lower bulk density of the LOX/HTPB leads to larger tank and chamber volumes, with associated increases in mass of these components relative to HP/HTPB.

Conclusions

A GA optimization methodology has been successfully applied to the design of a large hybrid rocket booster to minimize inert mass and to minimize GLOW. Results indicate excellent performance of the GA in this application, which contains both continuous variables (tank pressure, chamber pressure, and oxidizer massflux) and discrete variables (propellant combination and number of fuel ports). The capability to simultaneously handle both variable types is especially interesting in this context because the optimization of the number of fuel ports is one of the top-level considerations in hybrid booster sizing. Designs were generated without the expense or effort of an enumerative search strategy. Results indicate that the GA was able to generate optimal, or very nearly optimal designs in a space that exhibits a very broad, shallow minimum. Furthermore, the GA was able to find these solutions in the presence of noise arising from the weight convergence tolerance used in the HYROCS sizing code. The resulting designs are physically reasonable. The combination of a genetic algorithm with a rocket sizing code appears to offer benefits to the designers of rocket systems.

References

[1]Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Longman, Reading, MA, 1989.

[2]Lin, C.-Y., and Hajela, P., "Genetic Algorithms in Optimization Problems with Discrete and Integer Design Variables," *Engineering Optimization*, Vol. 19, No. 4, 1992, pp. 309–327.

[3]Crossley, W. A., and Laananen, D. H., "Conceptual Design of Helicopters via Genetic Algorithm," *Journal of Aircraft*, Vol. 33, No. 6, 1996, pp. 1062–1070.

[4]Heister, S. D., "Propellants for Low Cost Solid Rocket Boosters," The Aerospace Corp., TOR-0086(6453-01)-1, El Segundo, CA, Feb. 1986.

[5]Ventura, M., and Heister, S., "Hydrogen Peroxide as an Alternate Oxidizer for a Hybrid Rocket Strap-On Booster," AIAA Paper 93-2411, June 1993.

[6]Werthman, W. L., and Schroeder, C. A., "A Preliminary Design Code for Hybrid Propellant Rockets," AIAA Paper 94-0006, Jan. 1994.

[7]Vonderwell, D. J., Murray, I. F., and Heister, S. D., "Optimization of Hybrid Rocket Engine Fuel Grain Design," *Journal of Spacecraft and Rockets*, Vol. 32, No. 6, 1995, pp. 964–969.

[8]Heister, S. D., "A Computer Code for Solid Rocket Motor Design (SRMDES Version 2.0) Technical Description and User's Manual," The Aerospace Corp., TOR-0090(9975)-3, El Segundo, CA, June 1990.

[9]Kalt, S., and Bendal, D., "Conical Rocket Performance Under Flow Separated Conditions," *Journal of Spacecraft and Rockets*, Vol. 2, No. 3, 1965, pp. 447–449.

[10]Holland, J. H., *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA, 1992.

[11]Williams, E. A., and Crossley, W. A., "Empirically-Derived Population Size and Mutation Rate Guidelines for a Genetic Algorithm with Uniform Crossover," *Soft Computing in Engineering Design and Manufacturing*, edited by P. K. Chawdhry, R. Roy, and R. K. Pant, Springer-Verlag, Berlin, 1998, pp. 163–172.

J. A. Martin
Associate Editor