JOURNAL OF PROPULSION AND POWER
Vol. 15, No. 1, January–February 1999

Multidisciplinary Optimization in Conceptual Design of Mixed-Stream Turbofan Engines

Luc J. J. P. Nadon*
National Defense Headquarters, Ottawa, Ontario K1K 0K2, Canada
and
Stuart C. Kramer† and Paul I. King‡
U.S. Air Force Institute of Technology, Wright–Patterson Air Force Base, Ohio 45433-7765

Conceptual engine design is currently done essentially using a trial-and-error process based on the experience of engineers. This paper presents an alternative approach using multidisciplinary optimization to automate the conceptual design phase and ensure that an optimal design is reached. A robust integrated computer program was created to find the values of eight basic engine parameters that minimized fuel usage over a given mission. Because of the character of the objective function, a new approach was devised incorporating a genetic algorithm with a switch to a gradient-based algorithm to refine the design. The new procedure was used to optimize the engine design for a conceptual short-range interceptor. The process is general, however, and can be applied to a wide variety of missions and aircraft.

Nomenclature

C_{pAB} = specific heat at constant pressure in afterburner, Btu/lbm-R
C_{pc} = specific heat at constant pressure in compressor, Btu/lbm-R
C_{pt} = specific heat at constant pressure in turbine, Btu/lbm-R
C_{TO} = power takeoff shaft power coefficient
$e_{c'}$ = polytropic efficiency of fan
e_{cH} = polytropic efficiency of high-pressure compressor
e_{mH} = mechanical efficiency of high-pressure spool
e_{mL} = mechanical efficiency of low-pressure spool
e_{mPTO} = mechanical efficiency of power takeoff shaft
e_{tH} = polytropic efficiency of high-pressure turbine
e_{tL} = polytropic efficiency of low-pressure turbine
F = uninstalled thrust, lb
h_{PR} = heating value of fuel, Btu/lbm
h_t = specific enthalpy of turbine, Btu/lbm
M_5 = mixer core flow Mach number
$M_{5'}$ = mixer fan flow Mach number
m_0 = mass flow rate, lbm/s
$m_{0\mathrm{spec}}$ = specific flow, lbm/s-ft^2
N_{eng} = number of engines
P_{t3} = compressor exit total pressure, psi
P_{t5} = mixer core flow total pressure, psi
$P_{t5'}$ = mixer fan flow total pressure, psi
P_0/P_9 = nozzle exit plane pressure ratio
S = uninstalled specific fuel consumption (1/h), wing area, ft^2
T_{SL}/W_{TO} = thrust loading
T_{t3} = compressor exit total temperature, R
T_{t4} = high-pressure turbine inlet total temperature, R
T_{t7} = afterburner total temperature, R
W_A = avionics weight
W_{eng}/m_0 = specific engine weight, lb/lbm/s
W_f = fuel weight, lb
W_{fi} = initial fuel weight design variable, lb
W_P = payload weight, lb
W_{TO} = gross takeoff weight, lb
W_{TO}/S = wing loading, lb/ft^2
α = bypass ratio
β = bleed air fraction
γ_{AB} = ratio of specific heats in afterburner
$\Delta H_t/\theta$ = high-pressure turbine specific work, Btu/lbm
ε_1 = cooling air mass flow rate fraction for high-pressure turbine
ε_2 = cooling air mass flow rate fraction for mixer
η_{AB} = efficiency of afterburner
η_b = efficiency of burner
Π_{AB} = total pressure ratio of afterburner
Π_c = total pressure ratio of compressor
$\Pi_{c'}$ = total pressure ratio of fan
Π_{cH} = total pressure ratio of high-pressure compressor
$\Pi_{d,\max}$ = total pressure loss in diffuser (inlet) because of friction
$\Pi_{\mathrm{mix,max}}$ = maximum possible total pressure loss at mixer
Π_n = total pressure ratio of nozzle
Π_{tH} = total pressure ratio of high-pressure turbine
Π_{tL} = total pressure ratio of low-pressure turbine

Subscripts
max = maximum, with afterburner
mil = military power
min = minimum
SL = at sea level

I. Introduction

AIRCRAFT engine design and manufacturing is an extremely competitive field, involving design choices that impact the award of multimillion and even multibillion dollar contracts. Small variations in engine design parameters may have a significant impact on critical performance measures such as available thrust, fuel consumption over a mission, or engine and aircraft costs and, hence, on the eventual competitive position of the product. The current engine design procedure is largely a manual, trial-and-error process relying heavily on experience. This is partly because of the inherent

Received March 12, 1997; revision received May 20, 1998; accepted for publication June 23, 1998.

*Engineer, Store Clearances and Ballistics, Directorate of Technical Airworthiness 3-4-6, Major-General George R. Pearkes Building.

†Associate Professor, Department of Aeronautics and Astronautics, 2950 P Street, Building 640. E-mail: skramer@afit.af.mil.

‡Associate Professor, Department of Aeronautics and Astronautics, 2950 P Street, Building 640. E-mail: pking@afit.af.mil.

difficulty in predicting the off-design performance of sophisticated gas turbine engines, and partly because of the requirement to analyze the interactions between diverse factors such as three-dimensional viscous and turbulent flows, material stresses, and thermodynamics, to name a few. The objective of this paper is to present a unified, automated approach for producing optimized engine designs.

Optimization involves choosing appropriate measures to optimize. There are several engine characteristics that are suitable, such as fuel consumption, engine size, cost, and thrust. Other factors, such as producibility, maintainability, and service life, are influenced by choices in engine cycle parameters and sizing but are as yet difficult to introduce in conceptual design algorithms. A simple and effective objective in engine design is minimizing fuel usage because fuel consumption is a benchmark of engine efficiency and is a major component in long-term aircraft operating costs. Moreover, because fuel consumption is closely linked with other engine characteristics such as engine size and cost,[1] this study, as others before it,[2] focuses on the optimization of engine design for minimum mission fuel consumption. The procedure discussed next is general and can be used to optimize other objectives as well.

Conceptually, selection of the best engine for a given application simply requires evaluating the selected measure(s) of performance for all combinations of input variables, and then selecting the combination giving optimal performance while meeting mission and engine cycle constraints. Design variables are chosen for their relevance in evaluating the performance measure and include such parameters as compressor pressure ratios, turbine inlet temperature, and mixer Mach number. The range of design variables are further reduced to those combinations that do not violate any of the constraints or assumptions of all the analysis tools (such as empirical models, approximations, and experimental data) used to evaluate the engine performance. Even with reasonable discretization, however, the number of possible design combinations would still be far too large for an exhaustive search. The challenge is to rationally search through the set in a way that gives high confidence in finding the best combination, yet does not take too long to complete the effort.

A typical approach used in current practice is a manually iterative scheme. With a reference altitude and flight Mach number selected, a repetitive on-design cycle analysis is used to roughly define an initial feasible design region. One (or occasionally more) representative design is chosen from this region for off-design and mission analysis and, if needed, is modified by trial and error to determine a baseline design feasible for the entire mission. The mere act of finding a feasible baseline design can be a major effort. The baseline design is then improved upon by adjusting the engine design variables, usually one at a time, until no further reduction in, say, fuel consumption is obtained. The direction, magnitude, and possible combinations of these changes are based on sensitivity analysis, experience, and insight.

There are numerous problems associated with this method. First and foremost, it is time consuming as numerous on-design, off-design, and mission analyses must be performed by the designer. Even a single design point evaluation may require many analyses. For example, it is common in conceptual design for the aircraft fuel capacity to be undefined and dependent on the total engine fuel requirements for a nominal mission. To evaluate the mission performance of a given design, the designer must assume an initial fuel capacity to estimate the aircraft weight, which in turn is a factor in determining the mission fuel usage. The resulting fuel requirement may not be consistent with the original capacity assumption, in which case it must be modified and the analysis repeated until the mission requirement and initial capacity converge. Second, the approach relies heavily on heuristics and is strongly influenced by the experience and skill of the designers. Because of the complex interactions involved, the behaviors of engine cycle and mission analysis are not easy to predict, making it difficult to select design variable values. Finally, there is no way for the designer to establish whether the final design is, in any sense, an optimal design.

A desirable alternative to this approach is an integrated, automated tool that can perform engine optimization for a large number of design variables and over a wide range of aircraft types, missions, and flight conditions. This is an appropriate application for multidisciplinary optimization (MDO). MDO allows simultaneous consideration of factors in multiple disciplines such as thermodynamics, aerodynamics, material stresses, and cost analysis, and has been used successfully in several aerospace applications including the design of structures and control systems.[3]

A portion of MDO requires the application of standard optimization techniques. Familiar gradient-based methods such as simple steepest descent or the more sophisticated sequential quadratic programming (SQP) algorithm[4] are of limited use in this case for several reasons. First, because off-design and mission analyses typically give unreliable gradient information in the infeasible region, and because the initial design point is almost always not feasible, gradient methods often cannot get started. Gradient algorithms also tend to converge slowly when far from an optimum in highly nonlinear cases such as those considered next. Finally, these methods are only guaranteed to find a local optimum and are dependent on having a good starting point. Genetic algorithms (GAs), relatively new and perhaps unfamiliar to the engine designer, provide an alternative approach that does not require gradient information.[5] They converge fairly rapidly to the vicinity of the global optimum. GAs, however, have relatively coarse resolution and may require many function evaluations.

This paper describes an integrated, automated optimization tool developed by the authors. It is based on principles of MDO and exploits the strengths of both GA and SQP in a two-phase process. The tool includes mission analysis, on-design, and off-design cycle analysis programs created by the authors to compute the engine performance for a given design. The cycle analysis programs were designed to solve the thermodynamics of mixed-stream, low bypass turbofan engines because they are the predominant types in military aircraft applications. This feature does not, however, limit the use of MDO for other types of engines. As proof of this new MDO concept, the tool was used to generate the optimum engine design for a conceptual short-range interceptor previously investigated by the authors[1] using the manual, trial-and-error method referred to earlier.

II. Optimization Background

A. General Optimization Concepts

The purpose of optimization is to find the values of a set of independent (input) variables values that minimize (or maximize) the value of another functionally dependent (output) variable, perhaps subject to some limitations on the values of the independent variables or some other functionally dependent variables. The independent variables are the design variables and typically represent independently variable attributes of the item. The dependent variable is the objective and frequently is a critical performance measure. The relationship between them is the objective function, and may be an explicit equation or an implicit relation, such as the output of a complex computer model. The limitations are constraints and are imposed to ensure a design is within given resources and requirements.[4] Constraints may also be either explicit or implicit.

The standard optimization problem formulation is as follows.

Minimize:

$$f(\boldsymbol{x}) \equiv f(x_1, x_2, x_3, \ldots, x_n)$$

Subject to:

$$h_j(\boldsymbol{x}) = 0, \qquad j = 1 \dots p$$
$$g_i(\boldsymbol{x}) \le 0, \qquad i = 1 \dots m \tag{1}$$

where $f(\boldsymbol{x})$ is the objective function, x is a design variable vector of size n, $h_j(\boldsymbol{x})$ are p equality constraints, and $g_i(\boldsymbol{x})$ are m inequality constraints. All constraints are standardized to the preceding form to simplify optimization algorithms. A constraint of the form $a \le g(\boldsymbol{x}) \le b$ is a side constraint and is expressed in standard form by breaking it into two constraints, $-g(\boldsymbol{x}) + a \le 0$ and $g(\boldsymbol{x}) - b \le 0$. An inequality constraint is active if the value is at the limit, i.e., when $g_k(\boldsymbol{x}) = 0$.

B. Constraints and Penalty Functions

A common difficulty with optimization algorithms (including the GA described next) is an inability to deal with constraints, particularly if the constraints are implicit. The standard remedy for this situation is to use penalty functions to convert the problem to an equivalent unconstrained problem by modifying the objective function to obtain

$$F(\boldsymbol{x}) = f(\boldsymbol{x}) + r\sum_{j=1}^{p} H_j[h_j(\boldsymbol{x})] + r\sum_{j=1}^{m} G_j[g_j(\boldsymbol{x})] \tag{2}$$

where the penalty functions G_j and H_j return a positive value for undesirable values of the constraint and zero for desirable values.[4] For example, an appropriate penalty for equality constraints is $H(x) = x^2$. The total penalty for a given design is the sum of all the individual penalties. In general, the multiplier r may vary with the iteration; in this case, however, it was set to one.

C. MDO

The purpose of MDO is to find the optimum values for a set of design variables dependent on functions from various engineering disciplines, particularly when the output of one or more disciplines is required as input to one or more other disciplines and vice versa. The various disciplines may or may not share the entire set of design variables. Figure 1 summarizes the MDO process for a two-discipline problem. The output of each discipline's program, U_1 or U_2, feeds into the other program and into the optimizer. The optimizer, in turn, generates new values for the design variable vector, $\boldsymbol{x}_D$, used as input for both discipline programs. This process is repeated until an optimum design is reached that satisfies the requirements of both disciplines.[6] In the case of the engine optimization performed for this paper, the two fields of interest are thermodynamics (engine cycle analysis) and aerodynamics (mission analysis).

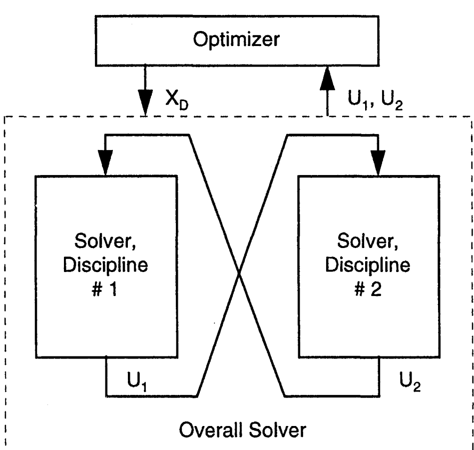

Fig. 1 Two-discipline MDO process.

D. GAs

GAs are relatively new in the world of optimization. As their name implies, GAs apply the biological principles of genetics and evolution to select an optimum design through a survival-of-the-fittest process spanning a number of generations created using reproduction, crossover, and mutation of design points.[5] They are very robust and have a number of advantages in complex problems. GAs are zero-order methods and do not use the derivative of the objective function, so they can easily handle sharp constraints, penalty functions, and nonsmooth objective functions. Because it explores most of the design space, a GA is very likely to find the global optimum. On the other hand, they can be computationally intensive, requiring many function evaluations. They also tend to converge more slowly as they get close to the optimum and have limited resolution so that the final result may need local refinement.

Basically, the GA provides a controlled method for randomly exploring the design space while retaining information on variable values that improve the objective function. A solution vector (chromosome) is represented by a binary string made from a concatenation of the binary representations of all of the design variables (genes). An initial population of chromosomes is created from randomly generated binary strings. The objective function is evaluated for each individual chromosome to select a reduced population surviving to the next generation. New populations evolve through chromosomal bit-swapping (crossover) and bit-flipping (mutation). This technique of bit manipulation also ensures that the algorithm does not bias itself toward a local minimum. Eventually, the population converges to a set clustered around a best solution, and the process is terminated when the population reaches a specified level of clustering. Because of the stochastic nature of the process and its limited resolution, it is unlikely that in every case the best solution will actually be the global optimum. This suggests using the GA to produce a good estimate of the optimum and then switching to a gradient-based optimizer to zero-in on the optimum.

E. SQP

SQP is an advanced nonlinear programming method that mimics Newton's method by using quasi-Newton methods to approximate the Hessian of the Lagrangian at each iteration of the optimization process.[4] A popular method to approximate the Hessian is the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method. These approximations are used to generate a quadratic subproblem (QP) which, in turn, is used to determine the search direction for a line search. The QP results from a quadratic approximation of the Lagrange function coupled with a local linearization of the nonlinear constraints and can be solved using any quadratic programming algorithm. One drawback of SQP methods is their requirement for a feasible starting point. This can be resolved with explicit constraints by solving a linear programming problem that minimizes the constraints' slack variables.[4] Once a feasible point has been located with the linear programming method, the algorithm may proceed with the SQP phase.

III. Engine Design Problem Formulation

A. Basic Problem

As mentioned earlier, the problem under consideration here is the conceptual design of an aircraft engine with minimum total fuel consumption over a nominal mission subject to constraints imposed by mission requirements and physical realizability of the design. We wish to do the following.

Minimize:

$$W_f = f(\boldsymbol{x}) \tag{3}$$

Subject to respecting limits on design variables, satisfying analysis assumptions, and meeting mission requirements. The objective function is the total fuel weight consumed for a given mission and is computed using the approach described in the next section. The primary engine parameters that control fuel consumption are relatively well understood[2] and form the basis for the vector of design variables for the problem:

$$x = [\Pi_c \; \Pi_{c'} \; T_{t4} \; T_{t7} \; \alpha \; m_0 \; C_{TO} \; M_5 \; W_{f_i}]^T \tag{4}$$

The only exceptional variable is the last one, initial (takeoff) fuel weight. The reason for its inclusion will be discussed later in this section. The constraints fall into two major categories, explicit limits on the design variables such as those imposed by technology, and implicit constraints such as those resulting from physical incompatibilities or from limitations in the fuel-consumption algorithm. As an example of the latter, if the engine-analysis program assumes a choked turbine (as the one used here did), any combination of parameters that leads to an unchoked condition will cause the program to fail and must be considered an infeasible point. Because there is no explicit relation that predicts such conditions, these constraints must be handled by penalty functions embedded in the analysis program.

B. Gross Weight Update

Inclusion of initial fuel weight as a design variable deserves some additional discussion. As noted earlier, mission analysis needs a takeoff weight to begin. The takeoff weight depends on the aircraft total fuel capacity and the aircraft structural weight, which itself is a function of the total fuel capacity. The total fuel capacity is generally chosen to provide a small excess over the total fuel required for the mission. The total fuel requirement, in turn, depends on the engine consumption and on the takeoff weight. This circularity results in the need, in manual calculations, to iterate each solution to achieve convergence of the weights. Iteration is avoided in this approach by the inclusion of initial fuel weight as a design variable and the addition of the constraint $W_{f_i} - W_f \geq 0$, which requires that there is enough fuel onboard to meet the mission requirements. (This constraint could easily be modified to enforce any excess fuel requirement.) Because fuel consumption increases with takeoff weight, the optimizer will find the minimum initial fuel weight compatible with the engine design and mission requirements.

C. Solution Approach

The objective function is highly nonlinear and occasionally mathematically ill-behaved. In particular, the results are unpredictable in the infeasible region, making it impossible to obtain reliable gradient information. This motivated the two-stage process alluded to earlier.

The overall optimization process is illustrated in Fig. 2. The first step of the optimization process consists of a search of the entire design space with a global optimizer. The purpose of this search is to identify feasible points and improve them to isolate the best candidate designs. A GA was selected to perform the global optimization because it does not require gradient information and is very robust. The second step of the process is local optimization using a gradient-based algorithm starting from the best feasible design provided by the global optimizer. The purpose of the local optimization is to efficiently refine the solution found by the GA. A BFGS gradient-based optimizer was used for the local optimization. If no feasible solutions were found, local optimization was not performed and the optimization process was stopped. As noted earlier, there are numerous and unpredictable ways a given design might fail a mission, such as lack of thrust or the failure of an engine to perform at any off-design condition. Because it cannot predict which variables to adjust or how to adjust them to meet the mission constraint, a gradient-based algorithm could not be relied upon outside the feasible region.

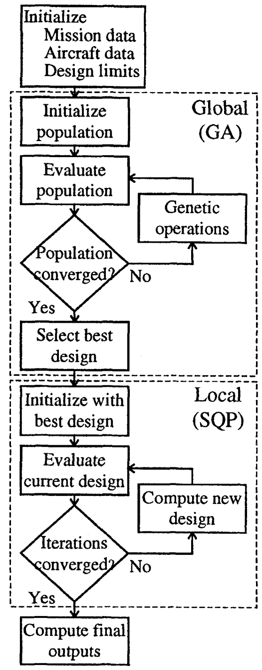

Fig. 2 Engine optimization.

IV. Optimization Program Components

The program developed as part of this study consisted of three analysis modules and two optimization modules controlled by a main program. The main program provides data interface and controls the switch from the global to the local optimizer. Computation of the objective function value W_f for a given design comes from on-design and off-design engine cycle analysis combined in the aircraft mission analysis. The analysis programs were made robust to ensure usable information was provided without interruption to the optimizer, even for highly infeasible designs. The following text is a brief overview of these modules.

A. Assumptions

The cycle and mission analysis programs described next were based on well-known assumptions and approximations used by Mattingly et al.[2,7] The most important assumptions are 1) both the low- and high-pressure turbines must remain choked; 2) component efficiencies are considered constant; 3) installation losses are set to a constant 9.1%; 4) the engine exhaust mixer area is constant; 5) the exit nozzle has a fixed throat; 6) basic aircraft data, such as the drag polar, are available; and 7) the reference flight condition for engine optimization is sea-level static.

B. On-Design Cycle Analysis Module

The purpose of on-design engine cycle analysis is to determine the engine parameters at the engine's reference flight condition (Mach number and altitude). In this project, the cycle of interest was the mixed stream, low bypass turbofan cycle.[2] Because on-design engine operation strictly occurs for only one flight condition—the reference condition—the engine will operate off-design for a significant part of a given mission (landing, climb, low-level flight, etc.). It is therefore of critical importance to select the on-design parameters to ensure acceptable performance over the entire aircraft flight envelope.

The cycle analysis will determine, among others, the values for uninstalled thrust, specific fuel consumption, mass flow, and specific thrust for a given set of engine parameters and flight conditions. The analysis consists of solving the simultaneous system of equations listed in Appendix E of Ref. 2, which evaluates the performance of each engine component. The on-design cycle analysis is critical as it is a part of both the off-design and mission analysis.

C. Off-Design Cycle Analysis Module

With a reference engine established, it is necessary, as part of the mission analysis, to evaluate the performance of the engine at other operating conditions. This is the role of the off-design cycle analysis. The model used in this study for off-design analysis is described in detail in Chap. 5 of Ref. 2.

D. Mission Analysis Module

The primary purpose of the mission analysis is to determine the total fuel usage W_f for an aircraft equipped with the reference engine over a complete mission from takeoff to landing. It is also used as an implicit constraint as part of the optimization process because engine designs that fail any part of the mission are rejected as infeasible. The analysis module considers each leg or legs of the mission, such as those listed next, sequentially and iterates with the off-design analysis module to determine the appropriate engine settings for that leg and the resulting fuel consumption for that leg. Mission analysis legs: warm-up, takeoff acceleration, takeoff rotation, constant speed climb, horizontal acceleration, climb and acceleration, loiter, constant altitude/speed cruise, constant altitude/speed turn, best cruise Mach and altitude, constant energy height maneuver, deliver expendables, and descent.

E. Global Optimization Module

The global optimization module is based on the GA realization, GAOT, developed at the North Carolina State University.[8] Some minor changes were made in the control and interface to integrate the module with the main program.

It was decided to allow the GA to run for five generations (five crossovers and mutations) with an initial population of 100 design variable sets (chromosomes). Many applications require more generations, but in this case the GA consistently found feasible designs within three generations.[8]

F. Local Optimization Module

The local optimizer is based on the Matlab implementation of the BFGS algorithm mentioned above.[9] Again, only minor changes were made to facilitate integrating this module with the main program.

V. Example: The Short-Range Interceptor

The short range interceptor (SRI) used to demonstrate the optimization process in this paper is a hypothetical concept previously investigated by the authors using the trial-and-error optimization process described earlier.[2] It is a small, high-thrust, highly maneuverable fighter aircraft with supercruise capability optimized for air combat. The SRI nominal mission profile is presented in Table 1. The aircraft and engine data for this aircraft are based on current technology for fighters and low-bypass, mixed-stream turbofans and are shown next.

1) *General data*: $N_{eng} = 1$, $T_{SL}/W_{TO} = 1.14$, $W_{TO}/S = 65$ lb/ft^2, $W_A = 1000$ lb, $W_P = 2634$ lb, $W_{eng}/\dot{m}_0 = 8$, and $\dot{m}_{0spec} = 37$ lbm/s-ft^2.

2) *Constraints*: $T_{t4} \leq 3200$ R, $T_{t7} \leq 3600$ R, $\dot{m}_0 \leq 270$ lbm/s, $0.2 \leq M_5 \leq 0.6$, $\Delta H_T/\theta \leq 32$ Btu/lbm, $0.2 \leq M_{5'} \leq 0.95$, $N \leq 110\%$, $T_{t3} \leq 1660$ R, $0.2 \leq \alpha \leq 1.0$, $3 \leq \Pi_{c'} \leq 5$, $24 \leq \Pi_c \leq 30$, $\Pi_{tL} \leq 0.5$, $P_{t5'} \geq P_{t5}$, and $P_{t3} \leq 375$ psi.

3) *Engine parameters*: $C_{pc} = 0.238$ Btu/lbm-R, $C_{pt} = 0.295$ Btu/lbm-R, $\gamma_c = 1.4$, $\gamma_t = 1.3$, $h_{pr} = 18{,}000$ Btu/lbm, $\varepsilon_1 = 0.05$, $\varepsilon_2 = 0.05$, $\Pi_b = 0.97$, $\Pi_{d,max} = 0.97$, $\Pi_n = 0.98$, $e_{c'} = 0.89$, $e_{cH} = 0.9$, $e_{tH} = 0.89$, $e_{tL} = 0.91$, $e_{mH} = 0.98$, $e_{mL} = 0.99$, $e_{mPTO} = 0.98$, $\eta_{AB} = 0.97$, $\gamma_{AB} = 1.3$, $C_{PAB} = 0.295$, $\Pi_{mix,max} = 0.97$, $P_0/P_9 = 1$, $\beta = 0.01$, $\Pi_{AB} = 0.96$, and $\eta_b = 0.98$.

The SRI and its mission were selected because it was a very constrained case due to the difficult mission requirements, such as the Mach 1.5 supercruise, the 5-*g* turn, the 1,500 ft short takeoff distance and the Mach 2.3, 50,000 ft maximum Mach/maximum altitude leg.

This case was considered ideal to put the two-step optimization process to the test as the restricted feasible design space and unpredictable mission analysis behavior make it difficult for any optimization algorithm to find a feasible design. It was noted when the SRI was investigated with the trial-and-error method that the main challenge for this aircraft was to find an engine design that could simultaneously perform the low-altitude loiter and the high Mach number/altitude leg.

VI. Results

The final design for the SRI case and its associated performance data are presented next.

1) *Best point from global optimizer*: $\Pi_c = 25.66$, $\Pi_{c'} = 3.68$, $T_{t4} = 2939$ R, $T_{t7} = 3576$ R, $\alpha = 0.69$, $\dot{m}_0 = 261.42$ lb/s, $C_{TO} = 0.0135$, $M_5 = 0.306$, and $W_f = 8{,}005$ lb.

2) *Final design from local optimizer*: $\Pi_c = 26.31$, $\Pi_{c'} = 4.0$, $T_{t4} = 2{,}927$ R, $T_{t7} = 3517$ R, $\alpha = 0.71$, $\dot{m}_0 = 265.0$ lb/s, $C_{TO} = 0.0127$, $M_5 = 0.30$, and $W_f = 7{,}950$ lb.

3) *Final performance*: $F_{SLmax} = 29{,}861$ lb, $F_{SLmil} = 18{,}385$ lb, $S_{max} = 1.769$ 1/h, $S_{mil} = 0.828$ 1/h, $F/\dot{m}_{0_{max}} = 112.68$ lbf/lbm/s, $F/\dot{m}_{0_{mil}} = 69.37$ lbf/lbm/s, $\Pi_{tL} = 0.394$, $\Pi_{tH} = 0.34$, $\Pi_{cH} = 6.58$, and $W_{TO} = 24{,}753$ lb.

It can be seen that the engine design and performance parameters are comparable to the performance of current state-of-the-art fighter engines. It was observed, as predicted in the preceding text, that the global optimizer's main hurdle was to locate a feasible design that could satisfy both the loiter and maximum Mach/maximum altitude legs. The first feasible design appeared in the second generation, and by the fifth generation nearly all of the population was in the feasible region.

Hand calculations showed that the feasible design region was quite small. Thus, the local optimizer could make little improvement. The objective function improvement was small,

Table 1 Short-range interceptor mission profile

Leg	Altitude, ft	Mach number	Comments
1. Warm up, taxi	2000	0	Idle
2. Takeoff in 1500 ft	2000	0.2	Max power
3. Accelerate to climb speed	0	0.2–0.7	Mil power
4. Climb and accelerate to cruise	0–30,000	0.7–0.9	Mil power
5. Accelerate to supercruise	30,000	0.9–1.5	Mil power
6. Supercruise 250 nm	30,000	1.5	Mil power
7. Launch AMRAAM	——	——	——
8. 360 deg, 5-*g* turn	30,000	1.6	Max power
9. 2 × 360 deg, 5-*g* turns	30,000	0.9	Max power
10. Combat acceleration	30,000	0.8–1.6	Max power
11. Fire AIM-9, gun	——	——	——
12. Escape dash 25 nm	30,000	1.5	Mil power
13. Subsonic cruise 200 nm	30,000	0.9	Part mil power
14. Loiter	30,000	0.41	Part mil power
15. Land in 1500 ft	2000	0.2	
Max Mach/max altitude	50,000	2.3	Max power

from 8005 to 7950 lb. Although the local optimizer was able to improve the best solution from the GA, it oscillated between the feasible and unfeasible mission region, showing that at least one of the implicit constraints was active. This small change indicates that the global optimizer was already close to the optimal design after only five generations.

The global optimizer behaved extremely well in the SRI case. It consistently converged to the best region in less than five generations. It was found to be advantageous to provide good estimates for reasonable design variable ranges to the GA (aided by engine parameters gleaned from historical data for similarly sized aircraft). This helped speed convergence to a feasible design, as highly infeasible designs would require much iteration and, thus, slow the optimization. Trial runs were often useful in determining suitable boundaries.

The local optimizer also behaved well, although in some other cases it would occasionally diverge when operating at the boundary of the feasible mission region. This was because a design may fail the mission, even after it was updated by the local optimizer, because of unreliable gradient information outside of the feasible region. Typical improvements in the objective function value during local optimization were less than 1%. This indicated that not only was the global optimizer very effective in finding good designs, but that for small feasible regions only small improvements can be found with a local optimizer. Nevertheless, the local optimizer is a needed addition, although in some cases it may only confirm the local optimality of the solution generated by the GA.

The program ran as expected with no user intervention between problem setup and examination of the results. Run times for the SRI case were about 10 h on a SPARC-10 workstation. Faster run times could be easily obtained by refining the program and compiling it, or by moving to a more capable workstation.

VII. Conclusions

The authors successfully developed an automated tool for optimizing low-bypass, mixed-stream turbofan aircraft engines during conceptual design. The process is not aircraft specific and can be applied to a wide range of missions and aircraft. This was accomplished by integrating engine cycle and mission analysis with GAs and SQP. The multidisciplinary optimization process ensured that all mission requirements and design limitations were met and that the final engine designs were the most fuel-efficient. Engine optimization for a short-range interceptor was investigated to illustrate the process.

The combination of a GA with a local optimizer proved very effective, with the GA exploring a given design space to define a restricted feasible region and the gradient-based optimizer quickly improving, whenever possible, the best solution provided by the GA. Constraints were successfully applied to the GA with the use of penalty functions. The programs developed were very robust, with engine analysis programs that consistently provided usable data to the optimizers, even for highly infeasible designs.

The authors believe that the tool described in this paper is a major advance, eliminating much of the manual groundwork and guesswork required in the early stages of conceptual engine design.

Acknowledgments

The authors would like to express their gratitude to Jeff M. Stricker at U.S. Air Force Wright Laboratory for his help with the problem definition, to Edward Pohl from the U.S. Air Force Institute of Technology for his optimization advice, and to Jack D. Mattingly from Seattle University for providing the original ONX, OFFX, and MISS Fortran computer programs. The views expressed are those of the authors and do not represent the official position of the U.S. Air Force, the Department of Defense, or the government of the United States, nor do they represent the position of the Canadian Armed Forces or the government of Canada.

References

[1]Nadon, L., "Multidisciplinary and Multiobjective Optimization in Conceptual Design for Mixed-Stream Turbofan Engines," M.S. Thesis, Dept of Aeronautics and Astronautics, U.S. Air Force Institute of Technology, Wright–Patterson AFB, OH, Dec. 1996.

[2]Mattingly, J., Heiser, W., and Daley, D., *Aircraft Engine Design*, AIAA Education Series, AIAA, New York, 1987.

[3]Sobieszczanski-Sobieski, J., and Haftka, R., "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments," AIAA Paper 96-0711, Jan. 1996.

[4]Arora, J., *Introduction to Optimum Design*, McGraw–Hill, New York, 1989.

[5]Goldberg, D., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison–Wesley, Reading, PA, 1989.

[6]Cramer, E., Dennis, J., Frank, P., Lewis, R., and Shubin, G., "Problem Formulation for Multidisciplinary Optimization," *SIAM Journal on Optimization*, Vol. 4, No. 4, 1994, pp. 754–776.

[7]Mattingly, J., *On-Design and Off-Design Aircraft Engine Cycle Analysis Programs*, AIAA Education Series, AIAA, Washington, DC, 1990.

[8]Houck, C., Joines, J., and Kay, M., "A Genetic Algorithm for Function Optimization: A Matlab Implementation," North Carolina State Univ., Raleigh, NC.

[9]Grace, A., *Optimization Toolbox User's Guide*, The MathWorks, Natick, MA, 1992.