JOURNAL OF PROPULSION AND POWER
Vol. 17, No. 1, January–February 2001

Optimization for Aircraft Engines with Regression and Neural-Network Analysis Approximators

Surya N. Patnaik*
Ohio Aerospace Institute, Brook Park, Ohio 44142-1068
and
James D. Guptill,† Dale A. Hopkins,‡ and Thomas M. Lavelle§
NASA John H. Glenn Research Center at Lewis Field, Cleveland, Ohio 44135-3191

The NASA Engine Performance Program (NEPP) can configure and analyze almost any type of gas-turbine engine that can be generated through the interconnection of a set of standard physical components. In addition, the code can optimize engine performance by changing adjustable variables under a set of constraints. For engine-cycle problems at certain operating points, the NEPP code can encounter difficulties: nonconvergence in Powell's optimization algorithm and deficiencies in the Newton–Raphson solver during engine balancing. A project was undertaken to correct the deficiencies. Nonconvergence was avoided through a cascade optimization strategy. Deficiencies associated with engine balancing were eliminated through neural-network and linear-regression methods. An approximation-interspersed cascade strategy was used to optimize engine operation over the flight envelope. Replacement of Powell's algorithm by the cascade strategy improved the optimization segment of the NEPP code. The performance of the linear-regression and neural-network methods as alternate engine analyzers was found to be satisfactory. The paper illustrates the results and insights gained from the improved version of the NEPP code considering two examples: a supersonic mixed-flow turbofan engine and a subsonic waverotor-topped engine.

Nomenclature

T_4 = temperature at entrance to high-pressure turbine
η = ratio of NEPP solution to cascade result

Introduction

THE NASA Engine Performance Program (NEPP) is a gas-turbine engine-cycle simulation code. This code can configure and analyze almost any type of gas-turbine engine that can be generated through the interconnection of a set of standard physical components: propellers, inlets, ducts, combustors, fans, compressors, turbines, shafts, heat exchangers, flow splitters, subsonic mixers and/or supersonic ejectors, nozzles and water injectors, or gas generators. The engine can be designed for different types of fuels: standard hydrocarbon jet fuel, cryogenic fuel, and slurries. For thermodynamic analysis built-in curve fits can be generated from empirical data available in NEPP. For the analysis of jet and rocket fuels, an auxiliary chemical equilibrium composition model is available.[1] The NEPP code has been successfully used to simulate a wide range of engines from turboshaft and turboprops to air-turbo rockets and supersonic variable-cycle engines. A description of the NEPP code, with typical input files for a set of engine configurations, is given in Refs. 2 and 3. Since its inception,[4] the NEPP code has been continuously undergoing improvement to keep pace with the advanced gas-turbine engines envisioned for the 21st century. NEPP simulation has decreased engine-cycle analysis time and improved model fidelity.

The NEPP code has a numerical optimization capability to increase engine performance. The program allows the optimization of a cost function for a set of independent variables subjected to a number of specified behavior parameters of the engine, which act as the constraints. In the NEPP code the resulting problem is solved using Powell's method,[5] which was developed in the early 1960s. The observation has been made that for certain engine problems Powell's method can produce an overdesign condition with fewer active constraints than the correct optimum solution or can experience convergence difficulties. A project was undertaken to correct the optimization-related deficiency in the NEPP code by augmenting it with a cascade optimization strategy that was developed recently.[6] During the implementation of the new algorithm for the solution of multimission engine problems with multiple operating points within a flight envelope, difficulties were encountered in engine-cycle analysis. To overcome the difficulties, approximate analyzers were created through two analysis approximators: neural-network and linear-regression methods. This paper summarizes the results and insights gained from the improved NEPP code.

The paper is organized as follows: an overview of the NEPP code with emphasis on engine operation optimization; two illustrative examples—a supersonic mixed-flow turbofan engine and a subsonic waverotor-topped engine; the cascade strategy with illustrations; a description of the neural-network and regression approximations; neural-network and regression solutions; and conclusions.

Overview of the Optimization Capability of NEPP

The NEPP code is popular in U.S. aircraft engine companies. The early versions of this code[4,7] have been improved by introducing multiple modes of operation to simulate variable-cycle engines, stacked component maps for variable geometry components and optimization capability. The current code[2] can simulate the steady-state design and off-design performance of almost any gas-turbine engine. Its chemical dissociation subprogram can model standard hydrocarbon jet fuel and cryogenic fuel and slurries. In addition to modeling airbreathing propulsion engines, NEPP can simulate air-turbo rockets, ejector mixers, and rockets.

The numerical optimization capability of the NEPP code, the subject matter of this paper, casts the engine operation as a standard nonlinear mathematical programming problem:

Find the design variables $\{D\}$;
To optimize a cost function C_f;

Received 31 March 1999; revision received 14 December 1999; accepted for publication 20 December 1999. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

*Engineer, 2280 Cedar Point Road; currently Senior Engineer, NASA John H. Glenn Research Center at Lewis Field, M/S 49-8, Materials and Structures, Cleveland, OH 44135-3191. Associate Fellow AIAA.
†Mathematician, Computational Sciences Branch, 21000 Brookpark Road.
‡Engineer, Machine Dynamics Branch, 21000 Brookpark Road. Senior Member AIAA.
§Engineer, Propulsion Systems Analysis Office, 21000 Brookpark Road.

Subjected to a set of inequality constraints $\{g\}$.

Engines have to perform satisfactorily over their flight envelopes, which consist of a number of operating points defined by altitude, Mach number, and power setting combinations. Engine operation design becomes a sequence of interdependent optimization subproblems, one for each operating point. The optimization process typically adjusts a few engine parameters. The difficulty in the optimization problem does not lie with the number of active design variables, but rather with its multiple operating-point character, constraint validity ranges, and the iterative nature of engine-cycle analysis. In the original NEPP code, subproblems were solved in sequence using Powell's algorithm. This algorithm has been replaced by a state-of-the-art cascade optimization strategy.

During engine optimization, an anomaly associated with the Newton–Raphson solver was observed in the analysis portion of the NEPP code. In the NEPP code an engine is balanced by varying independent parameters until dependent parameters are matched. For example, the compressor operating point must be varied until the compressor flow error is reduced to zero. In addition, the shaft speed must be varied until the amount of power being supplied by the turbine matches the amount of power being used by the compressor. The Newton–Raphson solver,[8] which does this balancing, uses the solution of the previous operating point as the initial guess for the current operating point. During an attempt to balance an engine, this solver converges for small changes in operating conditions or when the starting point is close to the solution point. The solver has trouble reaching convergence for relatively large jumps in operating conditions. In optimization such a deficiency becomes troublesome especially during one-dimensional searches, when the feasible design space can be violated. In most situations the algorithm returns to the feasible region to continue optimization calculations. However, the engine analyzer is unable to return to the feasible region when operating characteristics change abruptly, leading to a termination of the optimization process. The authors attempted to overcome this problem by approximating the engine analysis through neural-network and regression methods. An engine design optimization problem, in other words, can be solved using three analyzers: the original NEPP analyzer, the derived analyzer using neural-network approximations, and the linear-regression-based analyzer.

Behavior Parameters for Engine Optimization

NEPP provides a number of behavior parameters that can be used in engine operation optimization. In this paper only a partial list of possible design variables, cost functions, and constraints is provided.

Design Variables

The following parameters can be used as design variables for engine operation optimization:

1) D_1—rotational speed of the engine shaft that connects compressors, turbines, propellers, and other components;

2) D_2—tip speed of the propeller, when such a component is used in the engine;

3) D_3—waverotor speed, when such a component is used in the engine;

4) D_4—ratio of compressor bleed flow to total flow into the compressor;

5) D_5—compressor pressure ratio, its adiabatic efficiency, and its bleed (These parameters and the ratio of compressor bleed flow to total flow into the compressor can also be used for turbines and fans.);

6) D_6—three-dimensional map value for the fan;

7) D_7—nozzle flow area (i.e., exit area for convergent and throat area for convergent-divergent nozzles);

8) D_8—ratio of the exit-to-the-throat area for convergent-divergent nozzles;

9) D_9—geometrical nozzle parameters, such as length and divergence angle;

10) D_{10}—nozzle-exit static pressure;

11) D_{11}—geometrical parameters of the duct and burner;

12) D_{12}—ratio of the entrance and exit bleed flow to the total flow for the duct;

13) D_{13}—for splitters the bypass ratio in each branch;

14) D_{14}—primary flow temperature change in the heat exchanger;

15) D_{15}—heat added to the waverotor; and

16) D_{16}—primary and secondary flow area for mixers and ejectors.

Cost Function and Constraints

For operation optimization several engine parameters can be considered either as behavior constraints or as components of the cost function. Some behavior parameters that can form cost components follow:

1) C_1—Net engine thrust is a typical cost function. This parameter can be modified to account for drag from the installation, inlet, nozzle boattail, and other protuberances. The shaft and propeller horsepower are related cost components.

2) C_2—Fuel flow per unit time, or a combination of thrust and fuel flow (for example, the ratio of fuel flow to the engine thrust), can become a cost component. A constraint can also be specified on fuel flow.

3) C_3—The NO_x emission index (as the ratio of NO_x in grams to fuel in kilograms) can be a cost component. This variable can also become a constraint with a specified limit.

For optimization a merit function can be generated as a linear combination of the component costs as

$$C_f = \sum_k w_k \gamma_k \tag{1}$$

where C_f is the cost function, w_k is a weight factor, and γ_k is the cost for the kth component.

Behavior constraints can be imposed on the following engine parameters:

1) g_1—pressure ratio, temperature, and drag for inlets;

2) g_2—surge margin and pressure ratio for fans, compressors, and turbines;

3) g_3—exit temperature and corrected speed for fans and compressors;

4) g_4—flow factor for turbines;

5) g_5—jet velocity, overall pressure ratio, and static and critical pressure ratio at the throat for nozzles;

6) g_6—pressure, temperature, Mach number, and fuel-to-air ratio at specific flow stations along the engine;

7) g_7—mass-flow rate at the entrance or exit of the inlets;

8) g_8—bypass ratio, total-pressure losses, and total weight flow for flow splitters; and

9) g_9—total pressure drop, fuel mass flow, outlet temperature, and efficiency for burners.

The constraint set has to be modified for waverotor augmentation.

Illustrative Examples

Two engine problems are used to illustrate the cascade optimization strategy and analysis approximations. Engine 1 is a supersonic mixed-flow turbofan engine for the High-Speed Civil Transport system. Engine 2 is a subsonic waverotor-topped engine.

Engine 1: Supersonic Mixed-Flow Turbofan Engine

The supersonic mixed-flow turbofan (MFTF) engine with bypass flow is configured with 15 components. The components are mounted on two shafts with 17 flow stations: an inlet, a fan, a flow splitter, two ducts, a compressor, another duct, a burner, a high-pressure turbine, a low-pressure turbine, one more duct, a flow mixer, an afterburner nozzle, and so forth. The fan and low-pressure turbine are mounted on the first shaft. The second shaft carries the compressor, the high-pressure turbine, and a load. The engine was designed for a flight envelope with 122 operating points. The altitude of the operating points varies between sea level and 80,000 ft, while the speed changes between 0 and 2.4 Mach as shown in Fig. 1. The design objective at each operating point is to maximize the net thrust of the supersonic engine, accounting for the installation drag. The three active, independent design variables of the MFTF engine for these off-design points are

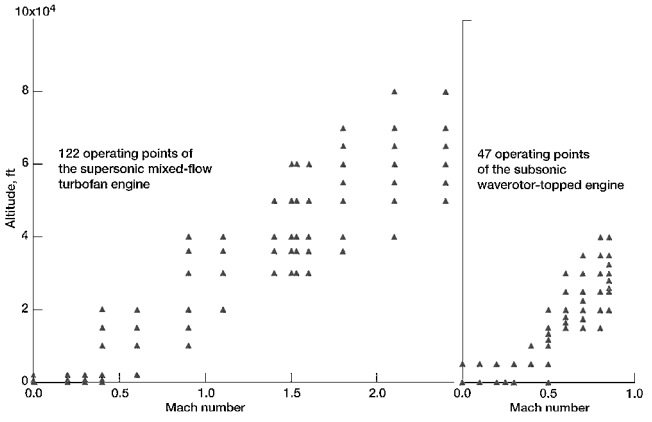


Fig. 1 Mission profile for supersonic and subsonic engines.

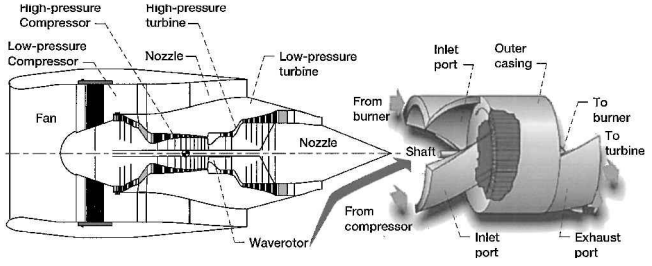


Fig. 2 Subsonic waverotor-topped gas-turbine engine.

1) D_1—bypass flow ratio of the splitter between flow stations 3 and 4 (flow station 3 is located downstream of the fan, and flow station 4 is located upstream of a duct leading to the compressor);

2) D_2—fan speed; and

3) D_3—fan surge margin.

Upper and lower bound constraints were specified on the following parameters:

1) g_1—inlet airflow ratio;

2) g_2—bypass ratio in the flow splitter;

3) g_3—burner exit temperature exclusive of nonheated air;

4) g_4—compressor exit temperature;

5) g_5—high-pressure turbine flow scale factor;

6) g_6—corrected speed ratio for the fan;

7) g_7—corrected speed ratio for the high-pressure compressor;

8) g_8—mixer primary entrance Mach number at flow station 11 (located upstream of a flow mixer leading through a duct to the nozzle);

9) g_9—mixer secondary entrance Mach number at flow station 16 (located upstream of the flow mixer);

10) g_{10}—R-value for the fan; and

11) g_{11}—R-value for the high-pressure compressor.

Engine 2: Subsonic Waverotor-Topped Engine

A high bypass-ratio subsonic waverotor-enhanced turbofan engine, depicted in Fig. 2, is considered as the second example. This engine is made of 16 components mounted on two shafts with 21 flow stations. It is modeled with an inlet and a splitter, which branch off to a compressor, a duct, and a nozzle. The main flow proceeds through a fan, a duct, a high-pressure compressor, another duct, a high-pressure turbine, a low-pressure turbine, one more duct, and a nozzle. The components mounted on the first shaft include the fan, the compressor along the secondary flow branch, the low-pressure turbine, and a load. The second shaft carries the high-pressure compressor along the main flow, the high-pressure turbine, and another load. A four-port waverotor (with its burner inlet and exhaust, compressor inlet, and turbine exhaust) is located between the high-pressure compressor and the high-pressure turbine. As shown in Fig. 1, the engine operating condition is specified by a flight envelope with 47 operating points, with altitudes between sea level and 40,000 ft, and speeds between Mach 0 and 0.85.

To examine the benefits that accrue from the waverotor enhancement, the engine is optimized under the assumption that most of the baseline variables and constraints are considered passive. The design objective is to maximize the net engine thrust, by considering the following two waverotor design variables as active:

1) D_1—heat added to the waverotor in the range 93,700–131,300 Btu/s; and

2) D_2—waverotor speed in the range 4940–7660 rpm.

An upper-and a lower-bound constraint were specified on each of the following engine parameters:

1) g_1—corrected speed ratio for the compressor along the secondary flow branch in the range 0.7–1.01;

2) g_2—corrected speed ratio for the fan in the range 0.7–1.01;

3) g_3—corrected speed ratio for the high-pressure compressor along the main flow in the range 0.7–1.01;

4) g_4—waverotor unmixed temperature in the range 2000–3200 °R;

5) g_5—surge margin on the compressor along the secondary flow branch in the range 15–100;

6) g_6—surge margin on the fan in the range 15–100;

7) g_7—surge margin on the high-pressure compressor along the main flow in the range 15–30; and

8) g_8—pressure ratio for the high-pressure turbine in the range 0.0–6.59.

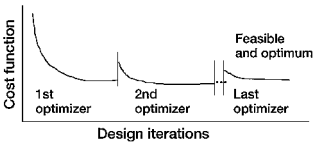
Cascade Optimization Strategy

Nonlinear programming optimization algorithms play an important role in the optimization of aircraft, their engines and structures, and other engineering problems. During the past few decades, several algorithms with associated computer codes have been developed. The performance of 10 different algorithms was evaluated for a set of 40 structural design problems with 10–50 design variables and a few hundred constraints. The observation was made that none of the algorithms could solve all of the problems, even though most algorithms succeeded in solving at least one-third of them.[9,10] The algorithms were used next to solve aircraft system design and variable-cycle engine problems. Even the most robust algorithms encountered difficulties in generating optimum solutions for the aircraft and engine problems. This can be attributed to diverse design variables and distorted design spaces. The aircraft problem, for example, required combining different types of design variables such as the wing area, engine thrust, temperature, and pressure ratio. The constraints (such as the takeoff and landing field lengths, compressor temperature, jet velocity, and climb thrust), which differed in magnitude and units of measure, distorted the feasible design space. The complexity was further compounded by the large sequence of implicit optimization subproblems that had to be solved for multiple operating points.

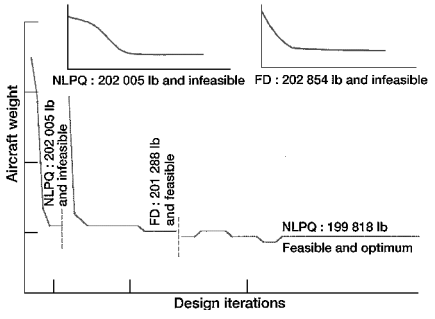
Improving the two key ingredients common to most algorithms—the search direction and step length—and thereby developing a superior optimizer was seriously considered, but ruled out because we believed that such aspects had been considered by the combined efforts of the developers of the algorithms.[11–15] Instead, a strategy[6] that would benefit from the strength of more than one optimizer was conceived. This strategy uses a number of optimization algorithms, one at a time, in a specified sequence (see Fig. 3a). The problem is solved by Optimizer 1. The second optimizer is begun from the Optimizer 1 solution with some pseudorandom damping, and so forth.

The cascade strategy is illustrated for a subsonic aircraft system design and the two engine problems: the MFTF engine (Engine 1) and the subsonic waverotor-topped engine (Engine 2). For the subsonic aircraft system design problem a three-optimizer cascade was used along with the aircraft system analyzer Flight Optimization System[16] of the NASA Langley Research Center. Aircraft weight was considered as the merit function. Design variables included the wing area, wing sweep, wing aspect ratio, wing taper ratio, wing thickness-to-chord ratio, and engine thrust. Behavior constraints were specified on the approach velocity, jet velocity, takeoff and landing field lengths, and missed-approach thrust.

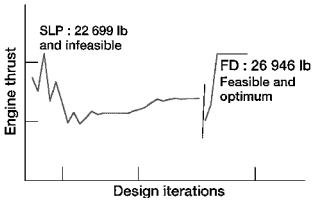
The cascade strategy used a sequence of three algorithms: nonlinear quadratic programming (NLPQ[11]), followed by the method



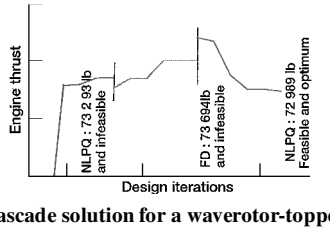
a) Cascade concept



b) Cascade and component solutions for a subsonic aircraft



c) Cascade solution for an MFTF engine



d) Cascade solution for a waverotor-topped engine

Fig. 3 Cascade solutions for aircraft and engines.

of feasible directions (FD[12]), followed by NLPQ again to solve the subsonic aircraft system design problem. Figure 3b depicts the cascade solution, along with solutions obtained from individual algorithms. The NLPQ method, used alone, converged to a heavy infeasible solution for the aircraft weight of 202,005 lb (see the insert in Fig. 3b). Likewise, the FD algorithm alone produced a heavy design of 202,854 lb (see the insert in Fig. 3b). In other words, neither the NLPQ nor the FD algorithm alone could successfully solve the subsonic aircraft system design problem. However, the NLPQ–FD–NLPQ cascade created from the same two optimizers successfully solved the problem with a feasible optimum solution at 199,818 lb for the aircraft weight (see Fig. 3b). The cascade strategy reduced the aircraft weight by 1.5%, eliminating infeasibility and overdesign conditions encountered by the individual cascade components. In a cascade (for example, NLPQ–FD–NLPQ) the same optimizer, NLPQ, can be used more than once, but we recommend that optimizer NLPQ be separated by another optimizer, such as FD.

The solution to the MFTF engine problem used a two-optimizer cascade: sequential linear programming (SLP[12]), followed by FD. This cascade algorithm converged to a feasible solution at 26,946 lb for the engine thrust (see Fig. 3c). The first algorithm, SLP, produced an infeasible underdesign at 22,699 lb for optimum thrust, and it violated 11 constraints. The second algorithm, FD, reached the optimum with a thrust value of 26,946 lb, which is feasible. The cascade strategy improved the engine thrust by 19% and eliminated design infeasibility.

The solution to the waverotor-enhanced subsonic engine (Fig. 3d) used a three-optimizer cascade: NLPQ followed by FD and NLPQ. The first NLPQ algorithm produced an infeasible design at 73,293 lb for the optimum thrust. The intermediate algorithm also produced an infeasible design at 73,694 lb. The final NLPQ algorithm reached the feasible optimum thrust at 72,989 lb. In summary, the cascade strategy successfully solved the subsonic aircraft problem and the two engine problems, even though its components encountered difficulties when used individually.

Solutions of the MFTF and Waverotor-Topped Engines

Solutions for the supersonic MFTF and subsonic waverotor-topped engines obtained using the cascade strategies and the original NEPP code (which used Powell's method) are depicted in Tables 1 and 2, respectively, for a few representative operating points. Figure 4 shows normalized solutions for both engines throughout their respective flight envelopes. The vertical axis in Fig. 4 represents an efficiency factor η (the ratio of NEPP solutions to the cascade results) for each operating point. A value less than unity for this factor ($\eta < 1$) represents superior performance for the cascade strategy. From Tables 1 and 2 and Fig. 4 the observation is made that the cascade solution produced higher thrust ($\eta < 1$) for most of the operating points. For a few operating points the two solutions agreed. The performance improvement can become significant if the design points with increased thrust are used to size the engine.

Engine Analysis and Optimization with Neural-Network and Regression Approximations

Linear regression and neural networks have been used to create approximate analyzers for engine analysis. The following steps were used in developing the approximate analyzers: 1) select the basis

Table 1 Solution for the 122-operating-point, supersonic MFTF engine

Operating points			Optimum thrust, lb		Cascade
Number	Altitude, ft	Mach number	Cascade solution	Original NEPP solution	improvement, %
20	Sea level	0.3	47,570	46,000	3.41
31	689	0.3	19,119	18,690	2.30
57	10,000	0.9	46,384	44,674	3.83
68	30,000	0.9	14,566	13,827	5.34
85	36,089	1.1	10,080	9,857	2.26
100	40,000	1.5	14,403	14,311	0.64
122	56,000	1.8	20,127	19,773	1.79

Table 2 Solution for the 47-operating-point, subsonic waverotor-topped engine

Operating points			Optimum thrust, lb		Cascade
Number	Altitude, ft	Mach number	Cascade solution	Original NEPP solution	improvement, %
1	Sea level	0.25	70,075	70,075	0.0
13	10,000	0.4	45,135	43,072	4.80
20	20,000	0.51	29,279	29,221	0.20
26	30,000	0.60	18,827	17,898	5.19
32	30,000	0.70	19,161	18,203	5.26
38	20,000	0.80	30,898	30,110	2.62
47	40,000	0.85	12,684	12,611	0.58

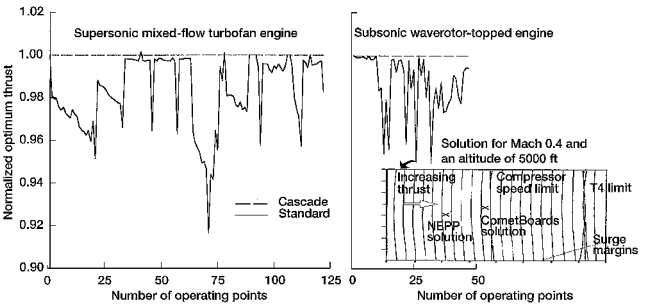


Fig. 4 Solutions for supersonic and subsonic engines, normalized with respect to cascade solutions.

functions; 2) establish a benchmark solution; 3) generate input-out pairs; 4) train the approximate methods; and 5) measure performance during optimization.

The MFTF and waverotor-topped engines required independent approximations for the cost function, as well as all of the constraints. For each method the MFTF engine with 22 constraints and 122 operating points required the development of 2806 independent functions. Likewise, the other engine required 799 functions for each method. Creating approximate models for engine problems requires the manipulation and management of an enormous amount of numerical data. To illustrate the approximate concepts in analysis and optimization while keeping numerical calculations within a manageable level, we selected the second problem—the waverotor-augmented subsonic engine with 10 operating points. For both the regression and neural-network methods, this problem only required the generation of 170 approximate functions. The regression and neural-network models are described next.

Linear-Regression Analysis

Linear-regression analysis can employ the following basis functions: 1) a full cubic polynomial, 2) a quadratic polynomial, 3) a linear polynomial in reciprocal variables, 4) a quadratic polynomial in reciprocal variables, and 5) combinations of these functions. The regression function for an n variable model with a cubic polynomial in the design variables and a quadratic polynomial in the reciprocal design variables has the following explicit form:

$$y(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \sum_{i=1}^{n}\sum_{j=i}^{n} \beta_{ij} x_i x_j + \sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=j}^{n} \beta_{ijk} x_i x_j x_k + \sum_{i=1}^{n} \bar{\beta}_i \frac{1}{x_i} + \sum_{i=1}^{n}\sum_{j=i}^{n} \bar{\beta}_{ij} \frac{1}{x_i x_j} \tag{2}$$

where y is the function to be approximated and $\boldsymbol{x}$ is the vector of independent variables. The regression coefficients $\{\beta\}$ are determined by using the linear least-squares method in the Lapack library[17] (DGELS routine). The gradient matrix of the regression function with respect to the design variables is

$$\nabla y = \left\{ \begin{array}{c} \dfrac{\partial}{\partial x_1} \\ \dfrac{\partial}{\partial x_2} \\ \vdots \\ \dfrac{\partial}{\partial x_n} \end{array} \right\} y \tag{3}$$

where

$$\frac{\partial y}{\partial x_\ell} = \beta_\ell + \sum_{i=1}^{n} \beta_{i\ell} x_i + \beta_{\ell\ell} x_\ell + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \beta_{ij\ell} x_i x_j + \sum_{i=1}^{n} \beta_{ii\ell} x_i^2 + \sum_{i=1}^{n} \beta_{i\ell\ell} x_i x_\ell + \beta_{\ell\ell\ell} x_\ell^2 - \frac{\bar{\beta}_\ell}{x_\ell^2} - \frac{1}{x_\ell^2} \sum_{i=1}^{n} \bar{\beta}_{i\ell} \frac{1}{x_i} - \frac{\bar{\beta}_{\ell\ell}}{x_\ell^3} \tag{4}$$

Once the regression coefficients have been obtained, the reanalysis and sensitivity analyses represented by Eqs. (2–4) require trivial computational effort.

Neural-Network Approximations

The neural-network approximation available for engine optimization is referred to as Cometnet. It approximates the function with a set of kernel functions:

$$y(\boldsymbol{x}) = \sum_{r=1}^{N}\sum_{i=1}^{n_r} w_{ri} \varphi_{ri}(\boldsymbol{x}) \tag{5a}$$

$$\frac{\partial y(\boldsymbol{x})}{\partial x_i} = \sum_{r=1}^{N}\sum_{i=1}^{n_r} w_{ri} \frac{\partial \varphi_{ri}(\boldsymbol{x})}{\partial x_i} \tag{5b}$$

where φ_{ri} represents the N kernel functions, n_r represents the number of basis functions in a given kernel, and w_{ri} represent the weight factors.

Cometnet permits approximations with linear, reciprocal, and polynomial, as well as Cauchy and Gaussian radial, functions. A singular-value decomposition algorithm[18] is used to calculate the weight factors in the approximate function during network training. A clustering algorithm in conjunction with a competing complexity-based regularizational gorithm[19] is used to select suitable parameters for defining the radial functions.

Generation of a Benchmark Solution for the Subsonic Waverotor-Topped Engine

Depending on the choice of the initial design, the nonlinear engine problem exhibited some variations in the optimum solutions at different operating points. To quantify the variation in the optimum thrust, the problem was solved using the original NEPP analyzer for 10 operating points, each using 10 different initial designs. The results are given in columns 4 and 5 of Table 3 and Fig. 5. The average solution for the 10 operating points is considered as the benchmark solution. The solution for thrust shows a maximum standard deviation of 1.2% for operating point 7 (see column 5 of Table 3). Modest variation was observed for the other nine operating points.

Table 3 Benchmark solution with standard deviation and optimum thrust for the waverotor-topped engine

Operating point			Benchmark solution of optimum thrust		Approximate solutions of optimum thrust, lb						
					Regular optimization		Cascade optimization solutions				
					NLPQ		FD–SUMT–NLPQ		NLPQ–FD–NLPQ		
Number	Altitude, ft	Mach number	Average solution, lb	Standard deviation	Regression	Neural network	Regression	Neural network	Regression	Neural network	NLPQ/NN–FD/ Reg-NLPQ/NEPP
1	0	0.25	70,030	83	69,980	70,094	69,980	70,094	69,980	70,094	70,165
2	0	0.10	80,220	48	80,188	80,271	80,188	80,271	80,188	80,271	80,238
3	0	0	88,827	55	88,794	88,818	88,794	88,818	88,794	88,818	88,751
4	5,000	0	73,641	31	83,307	83,454	73,631	73,647	73,631	73,647	73,633
5	0	0.10	80,029	377	80,207	80,289	80,207	80,289	80,207	80,289	80,318
6	5,000	0.10	66,941	106	75,232	63,291	66,893	66,911	66,897	66,890	66,901
7	0	0.20	72,963	899	73,052	73,156	73,052	73,156	73,052	73,156	73,079
8	5,000	0.20	61,703	101	68,357	67,675	61,647	61,664	61,645	61,664	61,641
9	0	0.30	69,209	370	69,165	69,213	69,164	69,174	69,167	69,213	69,184
10	5,000	0.30	57,446	61	62,348	62,206	57,450	57,472	57,450	57,472	57,437

Table 4 Good-quality input-output pairs for training the approximate methods

Operating point number	Number of good-quality input-output pairs	Bad quality input-output pairs			
		Zero thrust	Thrust difference exceeding 10 lb	Zero value for constraints	Constraint value difference exceeding 6%
1	2764	224	10	1	2
2	2768	223	7	2	1
3	2768	225	6	1	1
4	2764	235	2	0	0
5	2766	226	5	3	1
6	2763	237	0	1	0
7	2771	225	4	1	0
8	2763	235	0	1	2
9	2778	215	6	1	1
10	2772	227	2	0	0

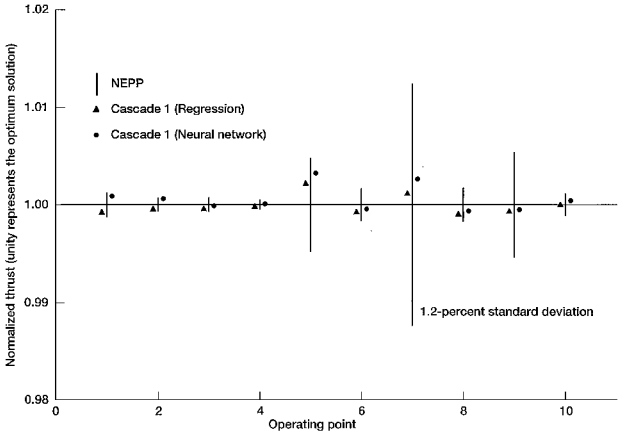


Fig. 5 Optimum solutions for a subsonic waverotor-topped engine at 10 operating points.

The benchmark solution is represented by the solid horizontal line in Fig. 5. The height of the vertical lines in Fig. 5 represents the standard deviation at each operating point.

Strategy to Generate the Input-Output Training Pairs

To generate the input-output pairs to train the approximate methods, 3000 design points were selected at random within their upper and lower bounds. The NEPP code was executed for the waverotor-topped engine analysis for 10 operating points for the 3000 design points. Midway through the first operating point, the NEPP analyzer encountered convergence difficulties—producing Not-A-Numbers followed by premature termination. For the 1500 points for which numerical solutions could be obtained, about 10%, or 150 cases, encountered convergence difficulty, producing zero-thrust conditions. For example, the design point with 112,350 Btu added and a speed of 3740 rpm produced a thrust of 56,805 lb. However, 103,080 Btu and 4826 rpm produced zero thrust instead of the correct thrust of 58,998 lb. Likewise, 113,000 Btu and 5505 rpm produced zero thrust instead of the correct 64,682 lb. Because it was suspected that this behavior was associated with the step size between neighboring design points, the 3000 design points were rearranged in an ascending order for each variable to obtain two sets of design points. Input-output pairs were generated for both sets. A point-by-point comparison was made between the output results obtained for the two sets. The following criteria were used to select good quality input-output pairs:

1) Pairs with zero-thrust conditions were excluded.
2) Pairs with thrust differences exceeding 10 lb were excluded.
3) Pairs with zero values for constraints were excluded.
4) Pairs with constraint differences exceeding 6% were excluded.

The selection process produced a set of good quality input-output pairs for the 10 operating points as shown in Table 4. On an average about 10% of the generated input-output pairs were excluded. The rejected pairs could have adversely affected the optimization process when the original NEPP analyzer was used.

Training the Approximate Methods

Both regression and neural-network methods were trained for the 10 operating points for the good-quality training pairs shown in Table 4. For regression approximations cubic polynomials were used in design variables, and quadratic polynomials were used in reciprocal design variables. Likewise, a Gaussian radial function was used to generate neural-network models for each operating point. For each of the 10 operating points, the process generated 17 regression and 17 neural network models (for 16 constraints and a cost function). A total of 340 approximate models were obtained for the 10 operating points. The quality of the reanalysis models was verified for a set of 100 random points by generating NEPP, regression, and neural-network solutions. The following error norms were used to quantify the level of accuracy.

Average error in the cost function for the neural-network method was defined as

$$\varepsilon_{\text{cf}} = \sum_{i=1}^{n} \left| \left(\frac{C_f^{\text{NN}} - C_f^{\text{NEPP}}}{C_f^{\text{NEPP}}} \right)_i \right| \Big/ n \tag{6a}$$

where ε_{cf} represents the average relative error, n is the total number of sample data points, and C_f^{NN} and C_f^{NEPP} are the values of the cost function from the neural-network calculation and the original NEPP methods, respectively.

Average error in a constraint for the neural-network method is defined as

$$\varepsilon_g = \sum_{i=1}^{n} \left| \left(g^{\text{NN}} - g^{\text{NEPP}} \right)_i \right| \Big/ n \tag{6b}$$

where ε_g represents the average error, n is the number of sample data points, and g^{NN} and g^{NEPP} are the values of the constraint from the neural-network calculation and the original NEPP methods, respectively. Because the constraints g represent normalized values, the error calculation is not further normalized. Likewise, average errors can be defined for the regression method.

Table 5 gives the error norm for a few typical engine parameters for both methods for each of the 10 operating points. For the neural-network method the maximum error in the thrust over the 10 operating points was about 0.25%. For the regression method the same parameters produced an error of less than 0.04%. Both neural-network and regression method performance for thrust estimation can be considered adequate. Over all of the constraints, the compressor surge margin exhibited the highest error, which was 1.25% for the regression method and 1.05% for the neural-network method. Both neural-network and regression methods trained satisfactorily for the subsonic waverotor-topped engine.

Optimization Using the Approximate Analyzers

The waverotor-topped engine was optimized using both neural-network and regression approximations, along with three different cascade strategies. The first cascade (FD–SUMT–NLPQ)

Table 5 Error norm in thrust and constraints for a waverotor-topped engine

Operating point	Variation in thrust, %		Auxiliary fan/compressor corrected speed (≤1.01)		Fan corrected speed (≤1.01)		Compressor corrected speed (≤1.01)		Burner outlet temperature (≤3200°R)		Auxiliary fan/compressor surge margin (≥15%)		Compressor surge margin (≥15%)	
	Regression	Neural network	Regression	Neural network	Regression	Neural network	Regression	Neural network	Regression	Neural network	Regression	Neural network	Regression	Neural network
1	0.033	0.25	0.012	0.107	0.012	0.107	0.027	0.034	0.037	0.08	0.073	0.478	1.238	0.994
2	0.036	0.23	0.017	0.108	0.017	0.108	0.032	0.034	0.041	0.084	0.069	0.426	1.169	1.034
3	0.036	0.219	0.018	0.109	0.018	0.109	0.032	0.036	0.041	0.085	0.069	0.419	1.127	0.996
4	0.015	0.18	0.013	0.116	0.013	0.116	0.048	0.051	0.022	0.103	0.044	0.369	1.059	1.049
5	0.037	0.23	0.017	0.108	0.017	0.108	0.032	0.034	0.041	0.084	0.071	0.426	1.171	1.034
6	0.017	0.191	0.014	0.116	0.014	0.116	0.05	0.051	0.022	0.101	0.046	0.373	1.041	1.033
7	0.034	0.241	0.013	0.107	0.013	0.107	0.031	0.034	0.039	0.082	0.069	0.45	1.236	1.049
8	0.018	0.205	0.017	0.115	0.017	0.115	0.051	0.053	0.023	0.097	0.053	0.406	0.993	0.908
9	0.032	0.254	0.012	0.105	0.012	0.105	0.024	0.034	0.036	0.076	0.074	0.499	1.184	0.937
10	0.025	0.223	0.022	0.116	0.022	0.116	0.043	0.049	0.028	0.09	0.074	0.473	0.989	0.775

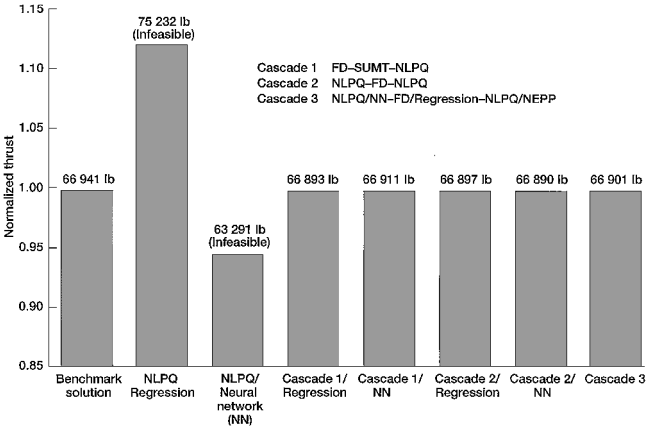


Fig. 6 Optimum thrust for a subsonic waverotor-topped engine at the sixth operating point.

used three algorithms in sequence: FD, the sequence of unconstrained minimizations technique (SUMT[15]), and NLPQ. The second cascade (NLPQ–FD–NLPQ) was created using two algorithms: NLPQ, FD, and NLPQ again. The third strategy is referred to as the approximation-interspersed cascade strategy (NLPQ/NN–FD/Regression-NLPQ/NEPP). Here, the engine was optimized first by the NLPQ algorithm with the neural-network analyzer, followed by optimization using the FD algorithm with the regression model, and finally, by the NLPQ algorithm with the original NEPP analyzer.

In Fig. 5 the optimum thrust obtained using the first cascade is represented by a triangle for regression and a circle for the neural network. The solutions obtained from both approximate methods lie within one standard deviation of the benchmark solution for each operating point. Both approximate methods performed satisfactorily.

For the sixth operating point optimum solutions obtained with and without the use of cascade strategies are depicted, along with the benchmark solution, in a bar chart (see Fig. 6). The solution for this operating point obtained using the approximate methods with no cascade strategy exhibited a variation of 12.4 and 5.5% for thrust. The solution also varied 16.9 and 5.8% for the added heat and 1.7 and 26.8% for the waverotor speed. The neural network with the NLPQ optimizer produced an infeasible design with a thrust that was 5.5% lower than the benchmark solution. Similarly, the regression model with the NLPQ optimizer produced an infeasible design with a thrust that was 12.4% higher than the benchmark solution. All three cascade solutions with the approximate analyzers agreed with the benchmark solution, with minor deviations. In other words, successful optimization of the subsonic waverotor-topped engine required the cascade strategy even for the neural-network and regression approximate methods.

CPU Time Estimation

The engine operation optimization is not computationally intensive. Furthermore, estimation of computation time has become less significant because the cost of computation has been continuously decreasing over the past few years. However, a time estimation is provided for completeness.

The numerical calculations reported in this paper used three different machines for convenience: International Business Machines RS6000 (for the generation of input-output pairs and for engine operation optimization), Silicon Graphics Indigo2 (for training the regression scheme), and Silicon Graphics Power Series 480–VGX (for training the neural network). The CPU times for the optimization using the original NEPP code with three different algorithms (NLPQ alone, Cascade 1, and Cascade 2) were 22, 85, and 116 s, respectively. The input-output pair generation to train the approximate analyzers consumed the bulk of the CPU time at 16,535 s (4 h, 35 min, 35 s). The time required does not account for the many unsuccessful attempts to generate good-quality input-output pairs. Regression training required a trivial amount of CPU time

at 14 s. The neural-network training time was 3914 s (1 h, 5 min, 14 s). Optimization run times with the regression analyzer for the three different cascade strategies were 4, 15, and 11 s, respectively. Optimization with the neural network required more time than the regression analyzer.

Conclusions

Replacement of Powell's optimization algorithm by the cascade strategy improved the performance of the optimization segment of the NEPP code. The maximum improvement in the thrust was 8% for the MFTF engine and 5% for the subsonic waverotor-toppedengine.

The performance of the linear-regression and neural-network methods as alternate engine analyzers was found to be satisfactory for the analysis and operation optimization of airbreathing propulsion engines. Both linear regression and neural networks performed at about the same level.

The engine optimization required the cascade strategy for solution using the original NEPP analyzer, as well as the neural-network and regression method-based analysis-approximators.

Airbreathing propulsion engines can be optimized using either the original NEPP code or an approximate analyzer and any of three methods: 1) a single state-of-the-art optimization algorithm, 2) cascade strategies, or 3) approximation-interspersed cascade strategies (i.e., a cascade strategy that includes approximate analyzers in the first few elements of the cascade and the original NEPP analyzer in the last element).

References

[1]Gordon, S., and McBride, B. J., "Computer Program for Calculation of Complex Chemical Equilibrium Compositions, Rocket Performance, Incident and Reflected Shocks, and Chapman-Jouguet Detonations," NASA SP-273, March 1976.

[2]Klann, J. L., and Snyder, C. A., "NEPP Programmers Manual," NASA TM-106575, Sept. 1994.

[3]Plencner, R. M., and Snyder, C. A., "The Navy/NASA Engine Program (NNEP89): A User's Manual," NASA TM-105186, Aug. 1991.

[4]Caddy, M. J., and Shapiro, S. R., "NEPCOMP—The Navy Engine Performance Computer Program, Version I," NADC-74045-30, April 1975.

[5]Powell, M. J. D., "An Efficient Method for Finding the Minimum of a Function of Several Variables Without Calculating Derivatives," *Computer Journal*, Vol. 7, No. 2, 1964.

[6]Patnaik, S. N., Coroneos, R. M., and Hopkins, D. A., "A Cascade Optimization Strategy for Solution of Difficult Design Problems," *International Journal for Numerical Methods in Engineering*, Vol. 40, No. 12, 1997, pp. 2257–2266.

[7]Fishbach, L. H., and Caddy, M. J., "NNEP—The Navy NASA Engine Program," NASA TM-X-71857, Dec. 1975.

[8]Pennington, R. H., *Introductory Computer Methods and Numerical Analysis*, 2nd ed., MacMillan, New York, 1970, p. 497.

[9]Patnaik, S. N., Coroneos, R. M., Guptill, J. D., and Hopkins, D. A., "Comparative Evaluation of Different Optimization Algorithms for Structural Design Applications," *International Journal for Numerical Methods in Engineering*, Vol. 39, Jan. 1996, pp. 1761–1774.

[10]Guptill, J. D., Coroneos, R. M., Patnaik, S. N., Hopkins, D. A., and Berke, L., "CometBoards Users Manual: Release 1.0," NASA TM-4537, Oct. 1996.

[11]Schittkowski, K., "NLPQL: A Fortran Subroutine for Solving Constrained Nonlinear Programming Problems," *Annals of Operations Research*, Vol. 5, No. 1–4, 1986, pp. 485–500.

[12]Vanderplaats, G. N., "DOT User's Manual, Version 2.00," VMA Engineering, Santa Barbara, CA, 1989, pp. E11–E35.

[13]Arora, J. S., "IDESIGN User's Manual Version 3.5.2," Optimal Design Lab., Univ. of Iowa, Iowa City, IA, 1989, pp. 1–101.

[14]"NAG Introductory Guide-MARK 16: E04UCF," NAG FORTRAN Library Routine Document, Downers Grove, IL, 1993, pp. E04:1–16.

[15]Miura, H., and Schmit, L. A., Jr., "NEWSUMT: A FORTRAN Program for Inequality Constrained Function Minimization, Users Guide," NASA CR-159070, June 1979.

[16]McCullers, L. A., "Aircraft Configuration Optimization Including Optimized Flight Profiles," *Recent Experiences in Multidisciplinary Analysis and Optimization*, edited by J. Sobieski, Pt. 1, NASA CP-2327, Jan. 1984.

[17]Anderson, E., Bai, Z., Bischof, C., Demmel, J., Dongarra, J., DuCroz, J., Greenbaum, A., Hammarling, S., McKenney, A., Ostrouchov, S., and Sorensen, D., *LAPACK User's Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992, pp. 12, 140, 141.

[18]Vetterling, W. T., Press, W., Teukolsky, S., and Flannery, B., *Numerical Recipes Example Book (C)*, Cambridge Univ. Press, New York, 1987, p. vii.

[19]Rissanen, J., "Stochastic Complexity," *Journal of the Royal Statistical Society, Series B—Methodological*, Vol. 49, No. 3, 1987, pp. 223–239.