AIAA-96-4021

6th AIAA/NASA/USAF Multidisciplinary Analysis & Optimization Symposium, Bellevue, WA, Sept. 1996

MULTI-LEVEL SIMULATION AND NUMERICAL OPTIMIZATION OF COMPLEX ENGINEERING DESIGNS

Mark Schwabacher* and **Andrew Gelsey**†
Rutgers, the State University of New Jersey
New Brunswick, NJ 08903

Abstract

Multi-level representations have been studied extensively by artificial intelligence researchers. We utilize the multi-level paradigm to attack the problem of performing multi-discipline engineering design optimization in the presence of many local optima. We use a multidisciplinary simulator at multiple levels of abstraction, paired with a multi-level search space. We tested the resulting system in the domain of conceptual design of supersonic transport aircraft, and found that using multi-level simulation and optimization can decrease the cost of design space search by **one or more orders of magnitude**.

Introduction

A major barrier to the use of gradient-based search methods for engineering design is that complex, multidisciplinary design spaces tend to have many apparent local optima — both real and pathological. We define an *apparent local optimum* to be a point that our optimizer, CFSQP [Lawrence *et al.* 1995], declares to be a local optimum. Such a point may be a true local optimum in the true physics of the problem, or it may be a *pathological local optimum.* Pathological local optima occur for several reasons. Numerical truncation error in the solvers within the simulator can result in local optima in the search space defined by the simulator that are not in fact local optima in the true physics. Discontinuities in the objective or constraint function or their derivatives — also known as "ridges" — violate CFSQP's assumptions and can therefore fool CFSQP into thinking that it is at a local optimum when in fact it is not. "Near ridges" — portions of the search space that are smooth, but that have very large second derivatives — can similarly fool CFSQP. Apparent local optima are a barrier to the use of optimization, whether they are real or pathological.

One approach to this problem is to use global search methods such as genetic algorithms and simulated annealing. We would, however, like to exploit the power of gradient-based optimization methods to quickly converge on the optimum. Our approach is therefore to use gradient-based optimization at multiple levels of search space abstraction (where each level has a much smaller number of apparent local optima), coupled with multiple levels of abstraction in the simulator.

Multi-level representations have been studied extensively by artificial intelligence researchers. Multi-level techniques for planning and theorem proving go back as far as [Sacerdoti 1974]. The importance of decomposing a problem into multiple levels was discussed at length in Simon's classic work on AI and Design, reprinted as [Simon 1981]. Some researchers have applied the multi-level paradigm to engineering design [Sobieszczanski-Sobiesky 1982, Sobieszczanski-Sobieski *et al.* 1985, Rogers 1989, Ellman and Schwabacher 1993], but have not focussed on the use of multi-level optimization to deal with multiple optima in the search space.

It might be asked how these problems are approached today. Human designers often decompose multidisciplinary design problems into smaller components that are then designed by different groups of people, but this entire process is generally carried out without the use of automated design. Our multi-level optimization method can therefore be

---

* Ph.D. 1996, Computer Science Dept. Now at the National Institute of Standards and Technology, Gaithersburg, MD. e-mail: mark.schwabacher@nist.gov

† Assistant Professor, Computer Science Dept. AIAA Member. e-mail: gelsey@cs.rutgers.edu

seen as an automation of the approach typically taken by groups of human designers.

We have tested our technique in the domain of conceptual design of supersonic aircraft, focusing on the airframe and the jet engine exhaust nozzle, and found that using multiple levels of simulation and optimization improves optimization performance by one or more orders of magnitude.

The Technique

Our approach is to use gradient-based optimization at multiple levels of search space abstraction, coupled with multiple levels of abstraction in the simulator. The search space abstractions are formed by decomposing the set of design parameters into two or more subsets. These subsets will typically correspond to different components of the artifact, such as the airframe and the nozzle of an aircraft. These subsets may be disjoint, or it may be necessary for a small number of design parameters to occur in more than one subset. The most important parameters of one component may be included in another component's abstract space in order to serve as a proxy for the first component during optimization. The search space abstraction should be defined in such a way that the different levels are as independent as possible — that is, the optimal values in one subset should not depend strongly on the values in the other subset. If the overall space is approximately a product space of the abstract spaces, and the abstract spaces each have a moderate number of local optima, then the number of local optima in the overall space will be approximately equal to the product of the number of optima in each of the abstract spaces. Thus by decomposing the problem into multiple levels, it should be possible to optimize in search spaces with a much smaller number of local optima. For example, it may be the case that the two decomposed spaces have n and m apparent local optima, respectively, and the overall space has mn apparent local optima. The number of multistarts needed to find the global optimum with a certain probability will vary linearly with the number of apparent local optima, so the cost of having a certain probability of finding the global optimum will be $O(mn)$ in the overall space, and only $O(m+n)$ in the decomposed space.

It is also helpful to have a simulator which can simulate at different levels of abstraction corresponding to the different levels of search space abstraction. When optimizing one component of the overall design, it helps to have a simulator that simulates the other components at a lower level of detail. Such a simplified simulator can be faster and less pathological than the full simulator.

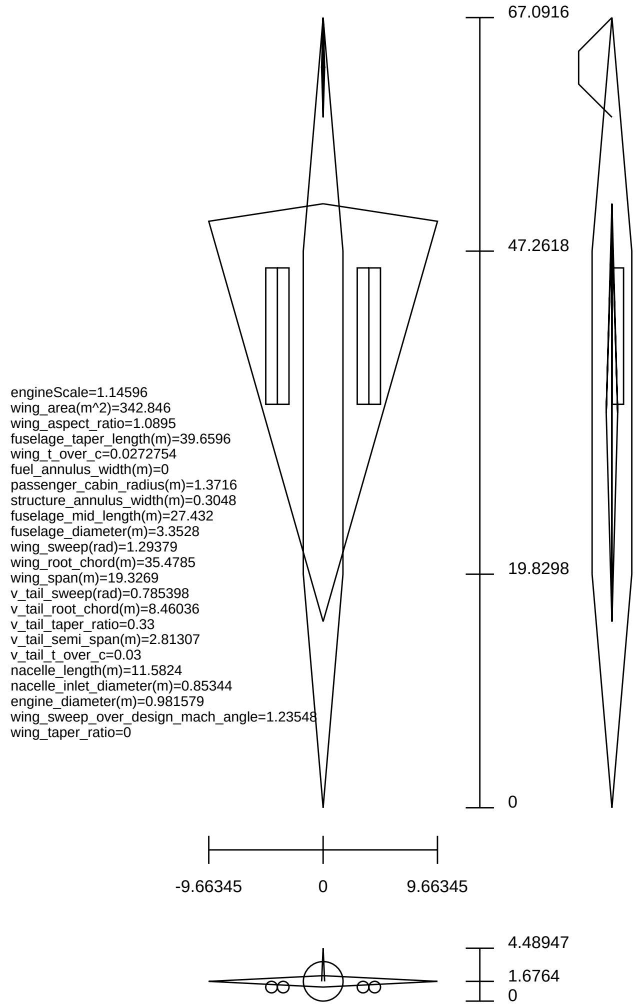

Figure 2: Supersonic transport aircraft designed by our system (dimensions in feet)

Aircraft Design

We have pursued our investigation in the domain of the design of supersonic transport aircraft [Gelsey and Smith 1996]. Figure 2 shows a diagram of the airframe of a typical airplane automatically designed by our software system to fly the mission shown in Figure 1. In our system, the optimizer attempts to find a good aircraft conceptual design for a particular mission by varying major aircraft parameters such as wing area, aspect ratio, engine size, etc. using a numerical optimization algorithm. The optimizer evaluates candidate designs using a multidisciplinary simulator. In our current implementation, the optimizer's goal is to minimize the takeoff mass of the aircraft, a measure of merit commonly used in the aircraft industry at the con-

Phase	Mach	Altitude (ft.)	Duration (min.s)	comment
1	0.227	0	5	"takeoff"
2	0.85	40,000	85	subsonic cruise (over land)
3	2.0	60,000	180	supersonic cruise (over ocean)

capacity: 70 passengers.

Figure 1: Mission specification for aircraft in Figure 2

ceptual design stage. Takeoff mass is the sum of fuel mass, which provides a rough approximation of the operating cost of the aircraft, and "dry" mass, which provides a rough approximation of the cost of building the aircraft. The simulator computes the takeoff mass of a particular aircraft design for a particular mission as follows:

1. Compute "dry" mass using historical data to estimate the weight of the aircraft as a function of the design parameters and passenger capacity required for the mission.

2. Compute the landing mass $m(t_{\mathbf{final}})$ which is the sum of the fuel reserve plus the "dry" mass.

3. Compute the takeoff mass by numerically solving the ordinary differential equation

$$\frac{dm}{dt} = f(m, t)$$

which indicates that the rate at which the mass of the aircraft changes is equal to the rate of fuel consumption, which in turn is a function of the current mass of the aircraft and the current time in the mission. At each time step, the simulator's aerodynamic model is used to compute the current drag, and the simulator's propulsion model is used to compute the fuel consumption required to generate the thrust which will compensate for the current drag.

To test experimentally the techniques described in this paper, we used a twelve-dimensional design space in which the optimizer varied the following aircraft design parameters over a continuous range of values:

1. engine size
2. wing area
3. wing aspect ratio
4. fuselage taper length (how "pointed" the fuselage is)
5. effective structural thickness over chord (a nondimensionalized measure of wing thickness)
6. wing sweep over design mach angle (a nondimensionalized measure of wing sweep)
7. wing taper ratio (wing tip chord divided by wing root chord)
8. fuel annulus width (the amount of space left in the fuselage for fuel)
9. nozzle convergent flap length (l_c)
10. nozzle divergent flap length (l_d)
11. nozzle external flap length (l_e)
12. nozzle radius 7 length (r_7)

Our optimizations focussed on two aspects of the aircraft: the airframe, which is described by the first eight parameters (see Figure 2), and the exhaust nozzle, which is described by the last four parameters.

Figure 3 shows the class of nozzles supported by the current system, the axisymmetric scheduled convergent-divergent exhaust nozzles often found in supersonic aircraft. [Mattingly *et al.* 1987] In Figure 3, r_{10}, r_e, and r_7 are fixed radii, and r_8 and r_9 are radii which are mechanically varied during aircraft operation. r_{10} is the outer radius of the engine to which the nozzle is attached, r_e is the radius of the duct leaving the engine, r_7 is the radius of the duct at the beginning of the movable convergent section of the nozzle, r_8 is the (variable) radius of the nozzle throat, and r_9 is the (variable) nozzle exit radius. Mechanically, this nozzle is a four-bar linkage, with three movable links labeled in Figure 3 by their lengths l_c, l_d, and l_e. During aircraft operation, the linkage is moved to change r_8 so that the cross-sectional area at the nozzle throat will produce desired engine performance. Since a four-bar linkage has one degree of freedom, setting r_8 also sets r_9. In the experiments described in this paper, we allowed the optimizer to vary l_c, l_d, l_e, and r_7.

Our aircraft simulator supports two different ways of simulating the nozzle. The first method takes as input the parameters describing the flap lengths within the nozzle, and simulates the actual operation of the nozzle throughout the mission. The second method uses what is known as

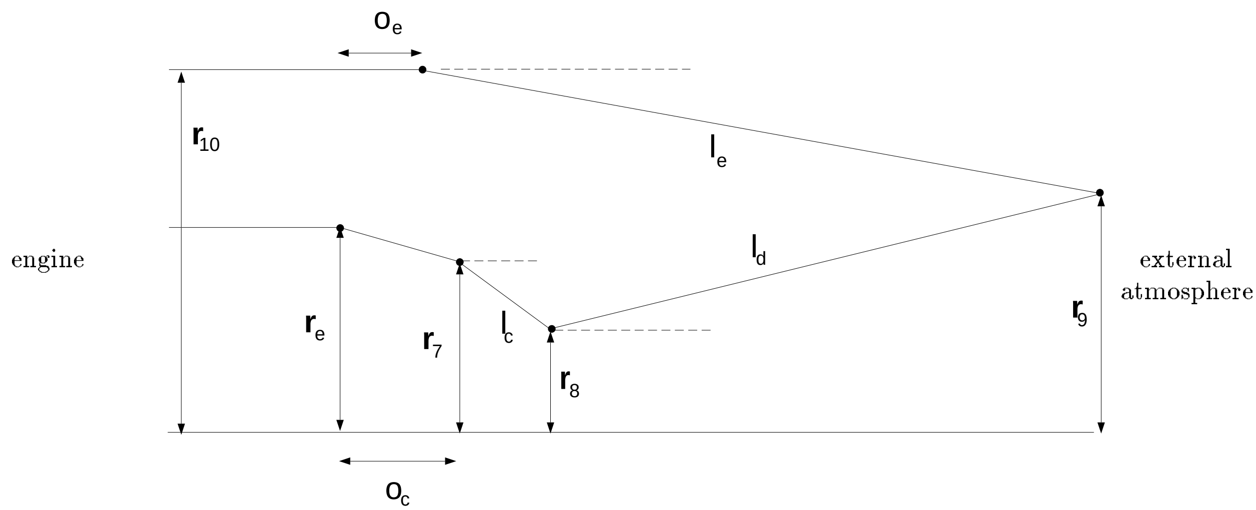

Figure 3: Axisymmetric convergent-divergent exhaust nozzle (flow from left to right)

an "ideal nozzle." This method does not actually simulate the movement of the flaps within the nozzle, but instead assumes that the nozzle will always produce a certain efficiency. This abstraction of the model allows faster simulation, and does not require the nozzle flap lengths to be input to the simulator. A complete mission simulation requires about 1/2 second of CPU time on a DEC Alpha 250 4/266 desktop workstation when using the four-bar nozzle, and about 1/4 second when using the ideal nozzle.

Search Procedure

In this paper we will focus on search of a space of candidate designs using numerical optimization methods which vary a set of continuous parameters to minimize[1] a nonlinear objective function subject to a set of nonlinear equality and inequality constraints. The numerical optimizer used in this paper is CFSQP [Lawrence *et al.* 1995], a state-of-the-art implementation of the Sequential Quadratic Programming method. Sequential Quadratic Programming is a quasi-Newton method that solves a nonlinear constrained optimization problem by fitting a sequence of quadratic programs[2] to it, and then solving each of these problems using a quadratic programming method. We have supplemented CFSQP with *rule-based gradients* [Schwabacher 1996] and with 35 *model constraints* [Gelsey *et al.* 1996].

The optimizations described in this paper were performed subject to a set of constraints. Figure 6 lists the constraints, along with the values of the constraint functions at the point which be believe is the global optimum. The nozzle geometry bound constraints require, for example, that the specified nozzle flaps be connectable. The table bound constraints require that the simulator not have to extrapolate outside the tables of experimental data which it uses. The aerodynamic bounds require, for example, that the lift coefficient required to fly the specified design over the specified mission not exceed one. There is a sanity check to make sure that the work performed by the actuators in the nozzles is positive. Finally, the passenger constraint requires that there be enough room in the plane for the specified number of passengers.

Experimental Results

We made the following hypotheses:

1. Using an appropriately selected two-level abstraction for optimization will produce better optimization performance (lower CPU cost for the same probability of getting a given design quality) than using one-level optimization.

2. Using an appropriately selected three-level abstraction for optimization will produce better optimization performance than using the two-level abstraction for optimization.

3. The same abstractions will produce good optimization performance for different missions.

To test our hypotheses, we performed optimizations using one, two, or three levels of abstraction, and then compared the results.

[1]To instead *maximize* the objective function, just multiply it by -1 and minimize.

[2]A quadratic program consists of a quadratic objective function to be optimized, and a set of linear constraints.

Design Parameter	low	high
engine size	0.5	3
wing area (sq. ft.)	1500	13500
wing aspect ratio	1	2
fuselage taper length (ft.)	100	200
effective structural thickness over chord	1	5
wing sweep over design mach angle	1	1.45
wing taper ratio	0	0.1
fuel annulus width (ft.)	0	4
nozzle convergent flap length (in.)	3	48
nozzle divergent flap length (in.)	9	120
nozzle external flap length (in.)	24	120
nozzle radius 7 length (in.)	1	100

Figure 4: Subset of design space explored

One-level optimization

First we tried doing optimization without the use of any multi-level techniques. We used the four-bar nozzle simulator, and used CFSQP to optimize in the search space defined by all twelve design parameters. Because this search space has many apparent local optima, we used a "random multistart" to attempt to find the global optimum. The system randomly generated starting points within the box of Figure 4 until it found 100 evaluable points[3], and then performed optimizations from each of these points. The first curve in Figure 5 shows the estimated cost of having a 99% chance of getting within various fractions of the point we believe to be the optimum using this method. This estimate is computed by multiplying the average cost per optimization times $\log(1 - P_{\text{desired}})/\log(1 - P_{\text{success}})$, where P_{desired} is the desired probability of getting within within the specified fraction of the apparent global optimum (99% in this case) and P_{success} is the probability of any single optimization getting within the specified fraction of the apparent global optimum (which we estimate using the fraction of our 100 optimizations that got within this fraction of the apparent global optimum)[4]

Two-level optimization

Since the one-level optimization was unacceptably expensive, we decided to attempt to reduce the optimization cost by decomposing the search space into two levels. As our first level, we used the eight airframe parameters, and the ideal nozzle simulator. As our second level, we used the four nozzle parameters, and the four-bar nozzle simulator. CFSQP quickly found the point that we believe to be the optimum in the eight-dimensional airframe space, which was encouraging. We then fixed the values of the eight parameters at their optimized values from the first level, and attempted to find the optimum in the nozzle space, using random multistart. After performing 1000 optimizations from random starting points, CFSQP failed to find even a single feasible point, so we declared this particular multi-level strategy to be a failure. We determined that the airframe designed in the first level, which had been designed using an ideal nozzle in the simulator, was only suitable for use with an ideal nozzle, so it was not possible to design a four-bar nozzle that would work with this airframe.

To circumvent this problem, we allowed CFSQP to vary all twelve design parameters in the second level. We performed a 5-point random multistart

[3]Some randomly generated designs, which we call "unevaluable points," cannot be simulated, either because the designs are meaningless or because of limitations of the simulator.

[4]Derivation of formula: $(1 - P_{\text{success}})$ is the probability that a single optimization will **not** find the global optimum, so $(1-P_{\text{success}})^n$ is the probability that **none** of n optimizations will find the global optimum, and thus $(1-(1-P_{\text{success}})^n)$ is the probability that at least one of n optimizations **will** find the global optimum. To find the cost of P_{desired}, a given desired probability of finding the global optimum, solve

$$P_{\text{desired}} = 1 - (1 - P_{\text{success}})^n$$

for n, which gives

$$n = \log(1 - P_{\text{desired}})/\log(1 - P_{\text{success}})$$

and finally multiply n by the the average cost per optimization. Note: the computed value of n is not necessarily an integer, so a more precise calculation would round n up to the nearest integer.

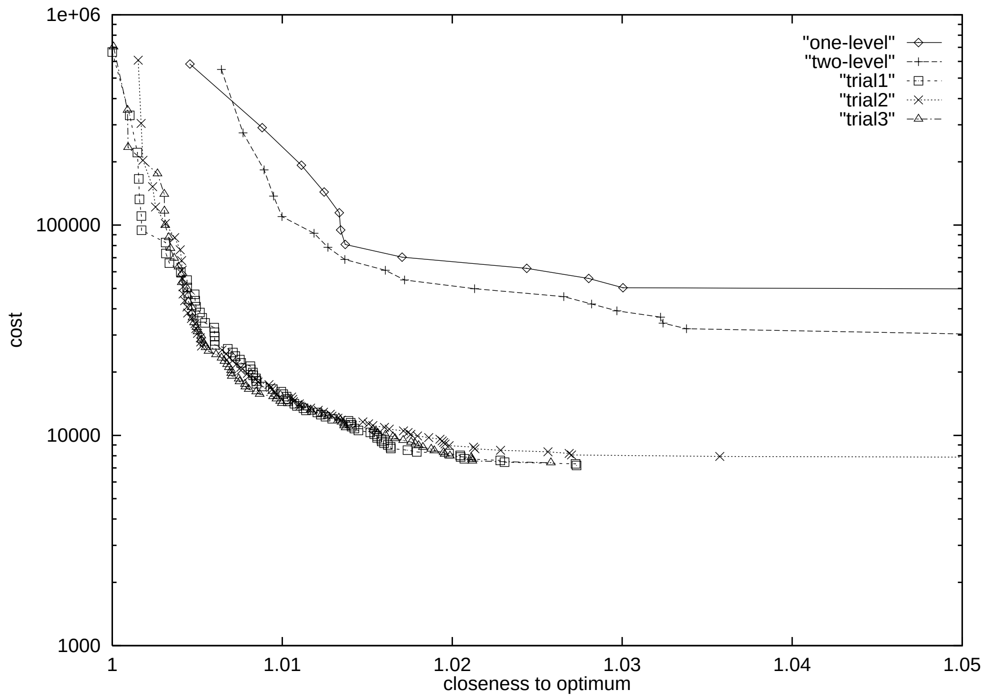

Figure 5: Performance of multilevel strategies for the first mission. Optimization performance increases as one moves down (lower cost) and to the left (closer to apparent optimum).

at the first (8-dimensional) level, followed by a 100-point random multistart optimization at the second (12-dimensional) level, where each optimization starting point had the eight airframe parameters set at their optimized values from the first level, and had randomly generated values of the four nozzle parameters. The second curve in Figure 5 shows the estimated cost of having a 99% chance of getting within various fractions of the apparent optimum using this method. Each point in this curve is based on the cost of doing the 5-point 8-dimensional multistart, plus an n-point 12-dimensional multistart, for a value of n corresponding to the cost.

Two-level optimization resulted in roughly a twofold reduction in the cost of getting within a certain distance of the apparent optimum, supporting our first hypothesis.

Three-level optimization

Two-level optimization significantly reduced the cost of finding the apparent optimum in the 12-dimensional airframe/nozzle space, confirming our first hypothesis. However, we believed that further improvements in optimization performance would be possible if we could allow CFSQP to optimize the nozzle without at the same time optimizing all of the airframe parameters. We decided to try a new strategy for the second level: letting CFSQP optimize the nozzle parameters, and just one airframe parameter. We chose wing area as the one airframe parameter to optimize in the second level, because we believe that it is the most important airframe parameter. One can think of wing area alone as an abstraction of the entire airframe, to be used while optimizing the nozzle, much as the ideal nozzle is used as an abstraction of the four-bar nozzle while optimizing the airframe. Each run at the second level started with the eight airframe parameters set to their optimized values from the first level, and with the the four nozzle parameters set randomly, and then did an optimization in the five-dimensional space defined by the four nozzle parameters and wing area.[5] We knew that optimizing in this space would not allow the optimizer to get exactly to the apparent global optimum, so we added a third level in which the optimizer is allowed to vary all twelve design parameters. The third level was run each time that the level two optimization ended at a feasible point. The three curves in Figure 5 labeled “trial1,” “trial2,” and “trial3” show the estimated cost of having a 99% chance of getting within various fractions of the apparent op-

[5]For each starting point, we kept wing area at its optimized value, rather than setting it to a random value, because we believe that the optimized value, although not optimal, should be better than a random value.

Design Parameters:	
engine size	1.462
wing area	4508 sq. ft.
wing aspect ratio	1.557
fuselage taper length	121.0 ft.
effective structural thickness over chord	2.978%
wing sweep over design mach angle	1.159
wing taper ratio	0.0
fuel annulus width	0.0
nozzle convergent flap length	15.31 in.
nozzle divergent flap length	70.54 in.
nozzle external flap length	101.48 in.
nozzle radius 7 length	14.63 in.
Objective Function:	
Takeoff Mass	162.2 tonnes
Nozzle geometry bounds:	
0.0-z7	-0.3971
r6-r10	-0.09784
r7-r10	-0.2806
z10-(z7+cl+dl)	-2.578
(r7-cl)-r6	-0.572
camin-camax	-1.272
el-elmax	-0.01806
elmin-el	-1.328
minRadius8-idealThroatRadius	-0.1240
idealThroatRadius-maxRadius8	-0.0001676
Table bounds:	
ECD lbte	-5.699
ECD ubte	-42.12
rae x min	-2.175
rae x max	-1.825
rae y min	-1.549
rae y max	-8.451
CA x min	-0.8136
CA x max	-98.03
CA y min	-4.121
CA y max	-35.71
CV x min	-0.8136
CV x max	-98.03
CV y min	-3.871
CV y max	-35.70
CB x min	-3.381
CB x max	-11.62
CB y max	-1.000
CB z min	-0.4876
CB z max	-0.5124
Aerodynamic bounds:	
wing-loading bound	-0.1497 tonnes
fuel mass constraint	0.0 tonnes
Lift coef-1	0.0
0.0-wing sweep	-1.214
wing sweep-pi/2	-0.3569
Sanity check:	
0.0-4barWork	-191015
Design Constraint:	
passenger constraint	-2

Figure 6: Best design found for mission of Figure 1. Negative values of constraint functions indicate that the constraints are satisfied.

timum using the three-level method. Each of these curves is based on a different 5-point multistart in the 8-dimensional space of level one, followed by an n-point 5-dimensional multistart (for various values of n corresponding to the costs) at level two, followed by a third level in which there is a 12-dimensional optimization from each of the feasible apparent optima of level two. We did three trials of the three-level method to see if the results would vary significantly based on what happens in the random multistart of level one; the graph in Figure 5 shows that there is not much variation.

Using the three-level method provided roughly an order of magnitude reduction in cost compared with the two-level method, confirming our second hypothesis. We believe that there are three reasons for this speedup. The first is that computing the gradient is less expensive in the five-dimensional space. The second is that in the two-level method, when CFSQP is started from a point in which eight of the design parameters are nearly optimal and the other four are set to random values, it does not know that the eight airframe parameters are near their globally optimal values, so it initially changes the airframe parameters to make the airframe more appropriate for the suboptimal nozzle, and later has to change them back as the nozzle becomes closer to optimal, resulting in the need to perform more line searches. The third reason is that when doing 5-dimensional optimizations, CFSQP has a higher success rate at finding a feasible point than when doing 12-dimensional optimizations. We believe that the reason for this higher success rate is that the constraint functions have fewer apparent local optima in the 5-dimensional space than they do in the 12-dimensional space.

Another mission

To test the effect of the mission on our results, we repeated the experiments for another mission — the mission of Figure 7. We compared the single-level method with the three-level method. The results are shown in Figure 8. The best design found for this mission is shown if Figure 9. We again obtained an order of magnitude reduction in cost using the multilevel method, confirming our third hypothesis.

Analysis

We believe that the full twelve-dimensional space has a large number of apparent local optima, so that finding the apparent global optimum requires a large number of random multistarts. The two-level strategy reduces the cost by getting eight of the twelve parameters close to their optimal val-

Phase	Mach	Altitude (ft.)	Duration (min.s)	comment
1	0.227	0	5	“takeoff”
2	0.85	40,000	50	subsonic cruise (over land)
3	2.0	60,000	225	supersonic cruise (over ocean)

capacity: 70 passengers.

Figure 7: Another mission specification

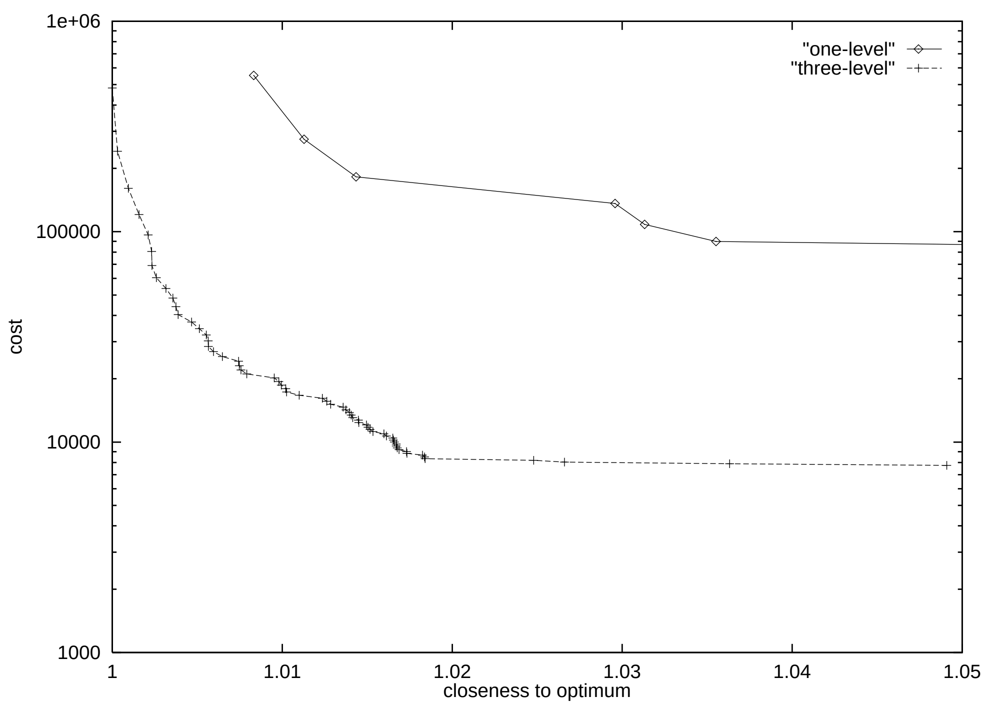

Figure 8: Performance of multilevel strategies for the second mission.

Design Parameters:	
engine size	1.155
wing area	3617 sq. ft.
wing aspect ratio	1.091
fuselage taper length	129.3 ft.
effective structural thickness over chord	2.673%
wing sweep over design mach angle	1.232
wing taper ratio	0.0
fuel annulus width	0.0
nozzle convergent flap length	28.25 in.
nozzle divergent flap length	50.34 in.
nozzle external flap length	94.40 in.
nozzle radius 7 length	14.89 in.
Objective Function:	
Takeoff Mass	132.1 tonnes
Nozzle geometry bounds:	
0.0-z7	-0.4095
r6-r10	-0.1053
r7-r10	-0.2196
z10-(z7+cl+dl)	-2.406
(r7-cl)-r6	-0.8320
camin-camax	-0.5552
el-elmax	-0.02223
elmin-el	-1.172
minRadius8-idealThroatRadius	-0.2032
idealThroatRadius-maxRadius8	0.0
Table bounds:	
ECD lbte	-14.46
ECD ubte	-9.046
rae x min	-1.789
rae x max	-2.211
rae y min	-2.005
rae y max	-7.995
CA x min	-0.9595
CA x max	-97.75
CA y min	-6.426
CA y max	-33.10
CV x min	-0.9595
CV x max	-97.75
CV y min	-6.176
CV y max	-33.10
CB x min	-1.634
CB x max	-11.98
CB y max	-1.000
CB z min	-0.5166
CB z max	-0.4834
Aerodynamic bounds:	
wing-loading bound	-0.1439 tonnes
fuel mass constraint	0.0 tonnes
Lift coef-1	0.0
0.0-wing sweep	-1.290
wing sweep-pi/2	-0.2803
Sanity check:	
0.0-4barWork	-233402
Design Constraint:	
passenger constraint	-2

Figure 9: Best design found for mission of Figure 7. Negative values of constraint functions indicate that the constraints are satisfied.

ues in Level 1, so that fewer random multistarts are needed in the twelve dimensional space. The three-level strategy provides a further improvement by getting all twelve parameters near their optimal values in Levels 1 and 2, so that fewer twelve-dimensional optimizations are needed in Level 3.

Limitations and Future Work

One might ask whether the multi-level technique is applicable to design problems outside the aircraft domain. We have formulated (but not yet tested) multi-level techniques for two other domains [Schwabacher 1996].

In the racing yacht design domain [Schwabacher *et al.* 1996], we could use two levels of representation and analysis for the keel. At the first level, the keel would be analyzed using the following simple algebraic formula for effective draft (where D is maximum draft, and A_{ms} is the cross-sectional area of the hull at mid-ship):

$$T_{eff} = 0.92\sqrt{D^2 - \frac{2A_{ms}}{\pi}}$$

At this level, the keel would be represented using a small set of parameters that have an effect on the formula, or on other quantities computed by the yacht simulator, such as surface area or displacement. This small set of parameters would include the keel's height and taper ratio.

At the second level, the keel would be analyzed using PMARC, a panel method. Since PMARC is sensitive to the shape of the keel, the keel would be represented using a B-Spline surface. The cost of analyzing the keel with PMARC is orders of magnitude greater than the cost of evaluating the algebraic formula, so it would be potentially very beneficial to perform most of the optimization at the first level.

In the supersonic missile inlet design domain [Zha *et al.* 1996], we have used an empirical code known as NIDA to analyze a missile inlet rapidly, and a computational fluid dynamics code known as GASP to analyze it with greater accuracy. Analyzing a single missile inlet with GASP takes about one CPU week, which makes it infeasible to perform optimizations with GASP using our current computational resources. We have instead performed optimizations with NIDA, and used GASP to verify the optimized designs. If we had greater computational resources available, we could perform inlet optimization at two levels, which would be likely to produce better designs than our current one-level NIDA optimization. The first level would be the same as our current optimizations — it would use

NIDA for the analysis, and a nine-parameter design space that allows the optimizer to vary only those aspects of the inlet that are properly modeled by NIDA. The second level would use GASP for the analysis, and would have a higher-dimensional design space (possibly using splines) that allows the optimizer to make a wider range of changes to the shape of the inlet, since GASP is much more sensitive to the inlet's shape.

It may be difficult to identify the appropriate simulator and search-space abstractions in still other domains. Automatically identifying such abstractions is an area for future research. Finally, the performance of our approach of performing optimization in the presence of many apparent local optima by using a gradient-based optimizer at multiple levels of abstraction needs to be compared with that of global methods such as genetic algorithms and simulated annealing. We may even find that it is possible to use these global methods at multiple levels of abstraction, for even better optimization performance.

Related work

Other work which uses the multi-level paradigm is described in the introduction section, but none of this work focuses on the problem of performing optimization in the presence of many local optima. A great deal of work has been done in the area of numerical optimization algorithms [Gill *et al.* 1981, Vanderplaats 1984, Peressini *et al.* 1988, Moré and Wright 1993, Papalambros and Wilde 1988], though not much has been published about the particular difficulties of attempting to optimize functions defined by large "real-world" numerical simulators. A number of research efforts have combined AI techniques with numerical optimization [Ellman *et al.* 1993, Schwabacher *et al.* 1994, Schwabacher *et al.* 1996, Tong *et al.* 1992, Powell 1990, Bouchard *et al.* 1988, Bouchard 1992, Agogino and Almgren 1987, Williams and Cagan 1994, Hoeltzel and Chieng 1987, Cerbone 1992], but have not addressed the issue of performing optimization at multiple levels.

Conclusion

Multi-level representations have been studied extensively by artificial intelligence researchers. We utilize the multi-level paradigm to attack the problem of performing multidiscipline engineering design optimization in the presence of many apparent local optima. We use a multidisciplinary simulator at multiple levels of abstraction, paired with a multi-level search space. We tested the resulting system in the domain of conceptual design of supersonic transport aircraft, and found that using multi-level simulation and optimization can decrease the cost of design space search by an order of magnitude.

Acknowledgments

We thank our aircraft design expert, Gene Bouchard of Lockheed, for his invaluable assistance in this research. We also thank all members of the HPCD project, especially Don Smith, Keith Miyake, Khaled Rasheed, and Tom Ellman. This research was partially supported by NASA under grant NAG2-817 and is also part of the Rutgers-based HPCD (Hypercomputing and Design) project supported by the Advanced Research Projects Agency of the Department of Defense through contract ARPA-DABT 63-93-C-0064.

References

A. M. Agogino and A. S. Almgren. Techniques for integrating qualitative reasoning and symbolic computing. *Engineering Optimization*, 12:117–135, 1987.

E. E. Bouchard, G. H. Kidwell, and J. E. Rogan. The application of artificial intelligence technology to aeronautical system design. In *AIAA/AHS/ASEE Aircraft Design Systems and Operations Meeting*, Atlanta, Georgia, September 1988. AIAA-88-4426.

E. E. Bouchard. Concepts for a future aircraft design environment. In *1992 Aerospace Design Conference*, Irvine, CA, February 1992. AIAA-92-1188.

G. Cerbone. Machine learning in engineering: Techniques to speed up numerical optimization. Technical Report 92-30-09, Oregon State University Department of Computer Science, 1992. Ph.D. Thesis.

T. Ellman and M. Schwabacher. Abstraction and decomposition in hillclimbing design optimization. Technical Report CAP-TR-14, Department of Computer Science, Rutgers University, January 1993.

T. Ellman, J. Keane, and M. Schwabacher. Intelligent model selection for hillclimbing search in computer-aided design. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, Washington, D.C., 1993.

Andrew Gelsey and Don Smith. Computational environment for exhaust nozzle design. *Journal of Aircraft*, 33(3):470–476, 1996.

Andrew Gelsey, Mark Schwabacher, and Don Smith. Using modeling knowledge to guide design space search. In J.S. Gero and F. Sudweeks, editors, *Artificial Intelligence in Design '96*, pages 367–385. Kluwer Academic Publishers, The Netherlands, 1996.

Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization*. Academic Press, London ; New York, 1981.

D. Hoeltzel and W. Chieng. Statistical machine learning for the cognitive selection of nonlinear programming algorithms in engineering design optimization. In *Advances in Design Automation*, Boston, MA, 1987.

C. Lawrence, J. Zhou, and A. Tits. User's guide for CFSQP version 2.3: A C code for solving (large scale) constrained nonlinear (minimax) optimization problems, generating iterates satisfying all inequality constraints. Technical Report TR-94-16r1, Institute for Systems Research, University of Maryland, August 1995.

Jack D. Mattingly, William H. Heiser, and Daniel H. Daley. *Aircraft Engine Design*. AIAA education series. American Institute of Aeronautics and Astronautics, New York, N.Y., 1987.

Jorge J. Moré and Stephen J. Wright. *Optimization Software Guide*. SIAM, Philadelphia, 1993.

P. Papalambros and J. Wilde. *Principles of Optimal Design*. Cambridge University Press, New York, NY, 1988.

Anthony L. Peressini, Francis E. Sullivan, and J. J. Uhl, Jr. *The Mathematics of Nonlinear Programming*. Springer-Verlag, New York, 1988.

D. Powell. Inter-GEN: A hybrid approach to engineering design optimization. Technical report, Rensselaer Polytechnic Institute Department of Computer Science, December 1990. Ph.D. Thesis.

J. L. Rogers. A knowledge-based tool for multilevel decomposition of a complex design problem. Technical Report NASA Technical Paper 2903, Langley Research Center, National Aeronautics and Space Administration, Hampton, VA, 1989.

E. D. Sacerdoti. Planning in a hierarchy of abstraction spaces. *Artificial Intelligence*, 5:115 – 135, 1974.

M. Schwabacher, H. Hirsh, and T. Ellman. Learning prototype-selection rules for case-based iterative design. In *Proceedings of the Tenth IEEE Conference on Artificial Intelligence for Applications*, San Antonio, Texas, 1994.

M. Schwabacher, T. Ellman, H. Hirsh, and G. Richter. Learning to choose a reformulation for numerical optimization of engineering designs. In J.S. Gero and F. Sudweeks, editors, *Artificial Intelligence in Design '96*, pages 447–462. Kluwer Academic Publishers, The Netherlands, 1996.

Mark Schwabacher. The use of artificial intelligence to improve the numerical optimization of complex engineering designs. Technical report, Rutgers University, October 1996. Ph.D. Thesis. http://www.cs.rutgers.edu/~schwabac/thesis.html.

H. Simon. *The Sciences of the Artificial*. MIT Press, Cambridge, MA, 1981.

J. Sobieszczanski-Sobieski, B. B. James, and A. R. Dovi. Structural optimization by multilevel decomposition. *AIAA Journal*, 23(11):1775–1782, November 1985.

J. Sobieszczanski-Sobiesky. A linear decomposition method for large optimization problems—blueprint for development. Technical Report NASA TM-83248, National Aeronautics and Space Administration, 1982.

Siu Shing Tong, David Powell, and Sanjay Goel. Integration of artificial intelligence and numerical optimization techniques for the design of complex aerospace systems. In *1992 Aerospace Design Conference*, Irvine, CA, February 1992. AIAA-92-1189.

Garret N. Vanderplaats. *Numerical Optimization Techniques for Engineering Design : With Applications*. McGraw-Hill, New York, 1984.

Brian C. Williams and Jonathan Cagan. Activity analysis: the qualitative analysis of stationary points for optimal reasoning. In *Proceedings, 12th National Conference on Artificial Intelligence*, pages 1217–1223, Seattle, Washington, August 1994.

G.-C. Zha, D. Smith, M. Schwabacher, A. Gelsey, and D. Knight. High performance supersonic missile inlet design using automated optimization. In *AIAA Symposium on Multidisciplinary Design*, 1996. AIAA Paper 96-4142.