Towards a Self-Configuring Optimization System for Spacecraft Design

Alex S. Fukunaga Andre Stechert Steve Chien

Jet Propulsion Laboratory, MS 525-3660
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109-8099
{alex.fukunaga, andre.stechert, steve.chien}@jpl.nasa.gov

Abstract

Spacecraft design optimization is a difficult problem, due to the complexity of optimization cost surfaces and the human expertise in optimization that is necessary in order to achieve good results. In this paper, wc propose the use of a set of generic, metaheuristic optimization algorithms (e.g., genetic algorithms, simulated annealing), which is configured for a particular optimization problem by an adaptive problem solver based on artificial intelligence and machine learning techniques. Wc describe work in progress on OASIS, a self-configuring optimization system based on these principles.

1 Introduction

Many engineering design optimization problems arc instances of constrained optimization problems. Given a set of decision variables X and a set of constraints C on X, the constrained optimization is the problem of assigning values to X to minimize or maximize an objective function $F(X)$, subject to the constraints C.

Although constrained optimization is a mature field that has been studied extensively by researcher, there arc a number of open, fundamental problems in the practical application of optimization techniques. In particular, the problem of selecting and configuring an optimization algorithm for an arbitrary problem is a significant obstacle to the application of state of the art algorithms to real world problems.

Wc arc currently developing the *Optimization Assistant (OASIS)* system, an automated, self-configurirrg optimization tool for spacecraft design these two issues. The goal of OASIS is to facilitate rapid "what-if" analysis of spacecraft design by developing a widely applicable, spacecraft design optimization system that maximizes the automation of the optimization process and minimizes the user effort required to configure the system for a particular optimization problem instance. In the rest of this paper, wc describe initial work on OASIS.

2 Resource-Bounded Black Box Optimization

The problem of global optimization on difficult, arbitrary cost surfaces is still poorly understood. The optimization of smooth, convex cost functions is well understood, and efficient algorithms for optimization on these surfaces have been developed. However, these traditional approaches often perform poorly on cost surfaces with many local optima, since they tend to get stuck on local optima. Unfortunately, many real-world optimization problems have such a "rugged" cost surface and arc thus difficult problems for traditional approaches to optimization.

In addition, many real-worlcl optimization problems arc black-lm *optimization proble$_m$) in* which the structure of the cost function is opaque. That is, it is not possible to directly analyze the cost surface by analytic means in. order to guide an optimization algorithm. For example, $F(X)$ can be computed by a complex simulation about which the optimization algorithm has no information (e.g., to evaluate a candidate spacecraft design, we could simulate its operations using legacy FORTRAN code about which very little is known to the optimizer except for its I/O specifications). Black-box optimization problems arc therefore challenging because currently known algorithms for black-box optimization are essentially "blind" search algorithms-instead of being guided by direct analysis of the cost surface, they must sample the cost surface in order to indirectly obtain useful information about the cost surface.

Recently, there has been much research activity in so-called *metaheuristic* algorithms such as simulated annealing [5], tabu search [2, 3] and genetic algorithms [4] for global optimization. These are loosely defined, "general-purpose" heuristics for optimization that proceed by iteratively sampling a cost sur-face, and they implement various mechanisms for escaping local optima, Although these algorithms have been shown to be successful on numerous applications with difficult cost surfaces, the behavior of these algorithms is still poorly un-

to nodes in the methogram correspond to input parameters for the component represented by the node, and outputs from a methogram node correspond to output values computed by the component. MIDAS is implemented as a CORBA object, and supports a wide variety of methods that can be used by external client systems (e.g., a GUI) to manipulate the methograms. This essentially provides an optimization system with a uniform interface for any design model encapsulate.d in MIDAS. Therefore, our solution to the problem of supporting a wide range of design models is to support an interface to MIDAS. That is, OASIS is designed to be an optimization system that can be used to optimize any MIDAS model.

Thus, the design model, which constitutes the user input to the OASIS system, is composed of the following:

. A M i DAS diagram that encapsulates the design model,

- A list of decision variables, as well as ranges of their possible values (may be continuous or discrete), and

. An output from a methogram node that corresponds the user's objective function value.[3]

Figure i shows part of a M i DAS methogram for the Neptune Orbiter model (see Section 4).

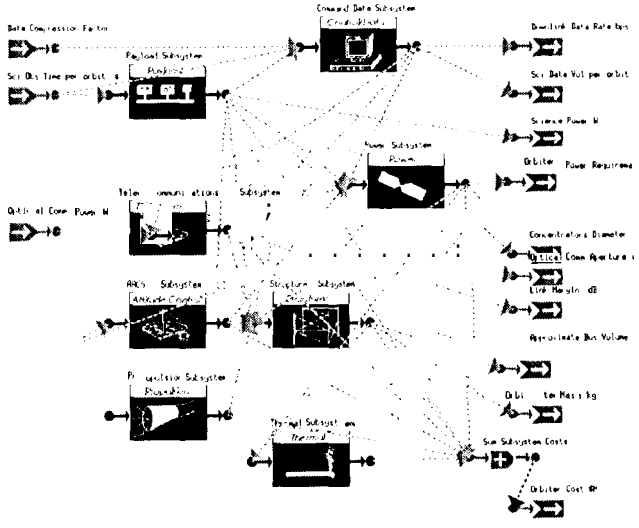

Figure 1: Screen shot of a MIDAS methogram (part of a Neptune Orbiter model).

3.2 Black-Box Optimization Algorithm Suite

OASIS includes a set of *configurable* ***black-box*** *optimization algorithms,* which are generic implementations of optimization algorithms that provide an interface for dynamic reconfiguration of their control points at runtime. Currently, this consists of a reconfigurable genetic algorithm [4] a reconfigurable simulated annealing algorithm [5], and some variants of traditional local optimization algorithms (Powel i's Method anti conjugate gradient algorithm) [6] with random restarts.

3.3 Self-Configuring Optimization System

Given a spacecraft design optimization problem instance in the form of a design model, the self-corrfiguriffg optimization component of OASIS selects and configures a metaheuristics from its suite in order to maximize some utility measure (usually, this is the quality of the design found by the system).

Our approach to optimizer configuration is to view it as a meta-level heuristic search through the space of possible problem solver configurations, where candidate configurations arc evaluated with respect to a utility measure, and the goal is find a configuration that maximizes this utility measure. In principle, it is possible to do a brute-force search through the space of possible problem solver configurations. This method is clearly intractable in general, since the number of configurations is exponential in the number of control points. Consider a problem solver with c control points, each with v values; there arc c^v problem solver configurations to bc considered.

Given the enormous computational expense of searching through the space of problem sol ver configurate ions, onc might wonder whether the search should/could be avoided altogether. To avoid search completely, there arc two alternatives. The first is to find a metaheuristic that outperforms all others for all problem instances (and thereby avoiding the problem of optimizer configuration altogether). As discussed in Section 2, we reject this solution as infeasible. The second alternative is a syntactic, "lookup-table" approach: classify the problem instance as a member of some class of problems, then apply the metaheuristic configuration that is known to work well for this class of problems. This method can work very wci i if we happen to have studied the class of problems to which the particular instance belongs, and we have available a good technique for classifying the instance as a member of the class. This approach, however, is of limited utility if wc encounter an instance of a class that we know nothing about, or if wc cannot correctly classify the problem as onc that belongs to a class for which wc have a good metaheuristic configuration.[4] l'bus, a purely syntactic approach does not suffice. A self-configuring optimization system needs to search the space of possible metaheuristic configurations-tile challenge is to discover and apply enough heuristic knowledge to the task to make it more tractable.

[3] The objective function could either be obtained directly from one of the existing outputs in the methogram, or it could be computed fry adding a new node that computes, e.g., a weighted linear combination of some set of output necks

[4] Indeed, the problems of defining useful notions of classes of problem instances, and classifying a problem instance as belonging to some particular class is a challenging pattern recognition problem in itself

4 Example Spacecraft Design Optimization Problems

In this section, wc describe two specific spacecraft design optimization problems to which wc arc currently applying the OASIS system. The first is a Iow-level optimization of tbc physical dimensions of a soil penetrator microprobe. The second is a system-level optimization of the configuration of the communication system of an orbiter spacecraft. These examples arc illustrative of the range of different optimization problems that arise in spacecraft design.

4.1 The Mars Soil Penetrator Microprobe

As part of' the NASA Ncw Millennium program, two microprobes, each consisting of a very low-mass aeroshell and penetrator system, arc planned to launch in January, i 999 (attached to the Mars Surveyor lander), to arrive at Mars in December, 1999. The probes will ballistically enter the Martian atmosphere and passively orient themselves to meet peak heating and impact requirements. Upon impacting the Martian surface, the probes will punch through the entry aeroshell and separate into a fore- and aftbody system. The forebody wi ll reach a depth of 0.5 to 2 meters, while the aftbody will remain on the surface for communications.

Each penetrator system includes a suite of highly miniaturized components needed for future micropenetrator networks: ultra low temperature batteries, power microelectronics, anti advanced micro-controller, a microtelecon munications system and a science payload package (a microlaser system for detecting subsurface water).

The optimization of physical design parameters for a soil penetrator based on these Mars microprobe is the first testbed for the OASIS system. The microprobe optimization domain in its entirety is very complex, involving a three-stage simulation:

- Separation analysis (i.e., separation from the Mars Surveyor),
- Aerodynamical simulation,
- Soil impact and penetration.

To illustrate the utility of adaptive problem solving, wc now briefly describe current work on a simplified version of the soil penetration stage.

Given a number of parameters describing the initial conditions including the angle of attack of the penetrator, the impact velocity, anti the hardness of the target surface, the optimization problem is to select the total length anti outer diameter of the penetrator, where the objective is to maximize the ratio of the depth of penetration to the length of the penetrator. We maximize this ratio, rather than simply maximizing tile depth of penetration, since for the Mars microprobe science mission, the depth of penetration should ideally penetrate at least the length of the entire penetrator).

Onc of tile initial condition parameters that has a significant impact on the structure of the cost surface for this optimization problem is the soil number, which indicates the hardness of the target surface. Intuitively, one would expect this to bc an important parameter, since, for example, it is clearly more difficult to penetrate harder targets (the penetrator could bounce off tbc target, for example).

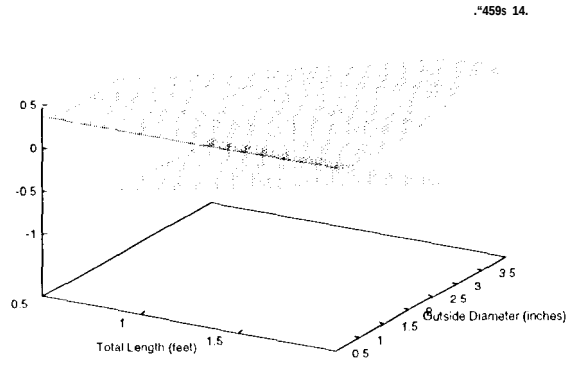

Figure 2: Sample points from cost surface for soil penetrator microprobe model. Plot of ratio of depth of penetration to length of penetrator. Soil number= 13 (soft soil). The z-axis represents the fitness value.

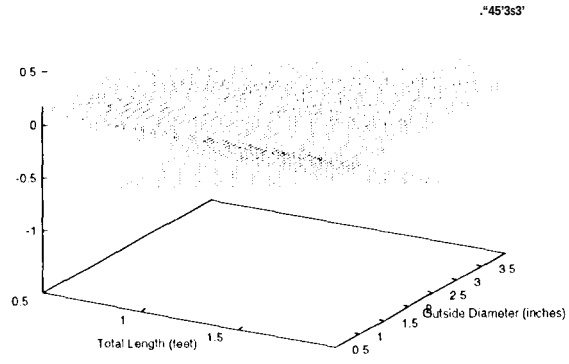

Figure 3: Sample points from cost surface for soil penetrator microprobe model. Plot of ratio of depth of penetration to length of penetrator. Soii number = 7 (hard soii).The z-axis represents the fitness value.

Figures 2 anti 3 show plots of sample points from the cost