SIAG/OPT Views-and-News

A Forum for the SIAM Activity Group on Optimization

Volume 17 Number 1 March 2006

Contents

Derivative Free and Surrogate Optimization
Introduction by the Editor
Luís N. Vicente 1
Nonlinear Programming by Mesh Adaptive Direct Searches
Mark A. Abramson, Charles Audet, and J.E. Dennis Jr. 2
Using Problem Structure in Derivative-Free Optimization
Philippe L. Toint 11
Space Mapping for Engineering Optimization
John W. Bandler, Slawomir Koziel, and Kaj Madsen 19

Bulletin 26
Chairman's Column
Kurt M. Anstreicher 29

Derivative Free and Surrogate Optimization

Introduction by the Editor

This issue of SIAM/Optimization Views-and-News focuses on Derivative Free and Surrogate Optimization. Hopefully, other issues in this area will appear in the near future, since a numbers of relevant topics were not covered yet.

I am extremely pleased to announce the three papers published now. In the first one, Abramson, Audet, and Dennis describe their mesh adaptative direct search (MADS) methods. These methods circumvent a drawback of the popular generalized pattern search (GPS) framework for nonsmooth problems. The usefulness of GPS and MADS partially relies on the explicit search/poll steps algorithmic framework.

Toint also covers direct and pattern search methods, but the goal of his paper is to show the use of partial separability in Derivative Free Optimization (DFO). In this paper, trust region interpolation based methods, a very efficient class of methods for DFO, is also considered.

The third paper is more along the lines of Surrogate Optimization (SO). Note that SO is revelant by itself and necessary also to make methods like MADS

or GPS more efficient in their use of the search step. The paper by Bandler, Koziel, and Madsen introduces to our audience the concept of space mapping, which allows to build useful surrogates when course models are available for fine tune models.

I would like to thank all the authors for their excellent contributions and to thank also Ana Luísa Custódio (New University of Lisbon) for reviewing some of the material in this issue.

Luís N. Vicente, January 2006.

Nonlinear Programming by Mesh Adaptive Direct Searches [1]

Mark A. Abramson
Air Force Institute of Technology, Department of Mathematics and Statistics, 2950 Hobson Way, Building 640, Wright Patterson AFB, Ohio, 45433 USA (Mark.Abramson@afit.edu).

Charles Audet
GERAD and Département de Mathématiques et de Génie Industriel, École Polytechnique de Montréal, C.P. 6079, Succ. Centre-ville, Montréal (Québec), H3C 3A7 Canada (Charles.Audet@gerad.ca).

J.E. Dennis Jr.
Department of Computational and Applied Mathematics, Rice University, MS 134, 6100 South Main Street, Houston, Texas, 77005-1892 USA (dennis@rice.edu).

Abstract This paper is intended not as a survey, but as an introduction to some ideas behind the class of mesh adaptive direct search (MADS) methods. Space limitations dictate a brief description of various key topics to be provided along with several references, which themselves provide further references.

The convergence theory for the methods presented here make a case for closing the gap between nonlinear optimizers and nonsmooth analysts. However these methods are certainly not of purely theoretical interest; they are successful on difficult practical problems. To encourage further use, we give references to available implementations. MADS is implemented in the direct search portion of the MathWorks MATLAB Genetic Algorithm and Direct Search (GADS) Toolbox.

1. Introduction – the problem and its properties

For us, derivative-free optimization excludes methods that use standard finite difference approximations to derivatives in a Newton or SQP algorithmic

[1] Audet's work was supported by FCAR grant NC72792 and NSERC grant 239436-05. Dennis was supported by LANL/DOE 94895-001-04 34, LBNL/DOE 671590, and both were supported by AFOSR F49620-01-1-0013, the Boeing Company, and ExxonMobil Upstream Research Company.

framework. Those are well established and valuable methods. Indeed, there are many reasons to use them in place of really derivative-free methods like the ones treated here if one can. However, our target class of problems are not amenable to such an approach.

Since this is to be only one of several papers devoted to derivative-free optimization, we will concentrate on summarizing our work without making an attempt to give a survey of the topic. This is a relief because derivative-free optimization is already a diverse area of optimization, and it is growing fast, due in part to the importance of these algorithms for applications.

Our interest in the topic of direct search methods came directly from users, and our interaction with users continues to be our strongest influence. However, it would be incorrect to assume that these methods are not interesting for their own sake. We have found all the theoretical challenge we would wish for in this area. This leads to another point we will try to make: derivative-free optimization will and should form closer ties between computational optimization and nonsmooth analysis. We believe that nonsmooth analysis should be included in any curriculum meant to train nonlinear optimization researchers.

In this paper, we consider the general nonlinear optimization problem:

$$\min_{x\in\Omega} \quad f(x) \tag{1}$$

where $\Omega = \{x \in X : c_j(x) \leq 0, j \in J\} \subset \mathbb{R}^n$ and $f, c_j : X \to \mathbb{R}^n \cup \{\infty\}$ for all $j \in J = \{1, 2, \ldots, m\}$, and where X is a subset of $\mathbb{R}^n$. No differentiability assumptions on the objective and constraints are required for these algorithms. However, the strength of the optimality results at a limit point are closely tied to the local smoothness of the functions there and to properties of the tangent cone to Ω at a limit point produced by the algorithm.

We treat X and $C(x) = (c_1(x), c_2(x), \ldots, c_m(x))^T \leq 0$ differently because they are intended to model different classes of constraints. The set X includes the set of points to which x must belong in order that the functions $f(x)$ and $C(x)$ can be evaluated. We only require that the user provides a routine that says whether or not x is in X. We refer to these constraints as "yes/no" or *oracular* constraints.

Another interesting aspect of these problems is that even when $x \in \Omega$, we may not get a value for $f(x)$ or $C(x)$, though it may take as long to find that out as it would have if we had been able to get a value. We model this situation by setting $f(x) = +\infty$. This happens, for example, in some multi-disciplinary optimization (MDO) problems, in which getting a value depends on runtime linking of legacy PDE solvers [17].

Every trial point, as well as the initial point, must satisfy the X constraints, but $C(x) \leq 0$ is only required to hold at the solution. In fact, the user is often interested in how much optimality is possible with a slight relaxation of these constraints. We call these *open* constraints, and we treat them by a modification given in [14] of the filter method.

Filter algorithms were introduced by Fletcher and Leyffer [31] as a way to globalize sequential linear and quadratic programming (SLP and SQP) without using any merit function for weighting the relative merits of improving feasibility and optimality. A filter algorithm introduces a function that aggregates constraint violations and then treats the resulting biobjective problem. A trial point is accepted if it either reduces the value of the objective function or that of the constraint violation; otherwise it is said to be filtered. Although this clearly is less parameter-dependent than a penalty function, specifying a constraint violation function still implies an assignment of relative weights to reducing the violation of each constraint. The algorithm maintains feasibility with respect to X by modifying the aggregate constraint violation for Ω to be $+\infty$ outside of X.

A key feature of the optimization problems we most often meet in practice is that they involve running an expensive simulation to get ancillary variables needed to evaluate the blackbox function codes that define f and C. This means that we need to be parsimonious with function and constraint values, and it also implies that there are likely to be few correct digits in the output. As a result, derivatives are unlikely to exist everywhere, and if they do exist, difference quotients are not likely to give derivative approximations suited to use in derivative-based algorithms.

Often in practice, users express a desire to obtain

the "global optimizer" of $f(x)$ on Ω. As we have described the problems, this is not something any algorithm can guarantee in practice. Still, global optimization algorithms generally find useful solutions when they can be applied to real problems. Indeed, with some attention to globality, the algorithms we outline here give equally useful solutions. We believe that this is because of synergy between this user request and another important user desire – robustness. In this context, one can think of a robust optimizer as one occurring in a broad valley. Such "global optimizers" are rather easier to find than those belonging to a narrow, but deeper basin.

2. What are mesh adaptive direct search (MADS) methods?

The methods we consider here are direct search methods. As the name *mesh adaptive direct search* (MADS) implies, these methods generate iterates on a tower of underlying meshes on the domain space. However, also as the name implies, they perform an adaptive search on the meshes including controlling the refinement of the meshes. The reader interested in the rather technical details should read [12, 13]. Here we ask the reader to imagine an underlying mesh and an algorithm for generating trial points on the mesh and adapting the fineness of the mesh to approach a local optimizer. We stress that the full mesh is never explicitly generated.

It is possible to dispense with the mesh as in [43, 44], which seems a simplification on the face of it. The argument against doing away with the mesh is that one must then use a sufficient decrease condition rather than accepting any point that provides simple decrease. Sufficient decrease conditions in the derivative-free situation are not as simple as a backtracking Goldstein-Armijo strategy in the quasi-Newton case [30]. Our suspicion is that whether or not to use the mesh is a matter of taste, not of algorithmic effectiveness, though we have no actual experience without the mesh on real problems.

Above, we mentioned the utility of nonsmooth analysis in derivative-free optimization. MADS is a case in point. We discovered MADS as a direct result of weaknesses in the generalized pattern search (GPS) class of algorithms [47], when applied to nonsmooth problems, which were exposed when we used nonsmooth analysis to analyze GPS [12, 14].

We could also call the methods considered SEARCH – POLL methods because each iteration consists of two steps, SEARCH and POLL . The goal of an iteration is to find unfiltered points in X. If SEARCH fails to find an unfiltered point, then POLL is executed, and if POLL does not succeed, then the mesh is refined.

The SEARCH step is crucial in practice because it is so flexible, but it is a difficulty for the theory for the same reason. SEARCH can return any point on the underlying mesh, but of course, it is trying to identify an unfiltered point. Any aspirations to find a local minimizer in a deeper basin than the one we start in is concentrated in the SEARCH step. When we discuss some SEARCH strategies, we will justify this point.

The POLL step is more rigidly defined, though there is still some flexibility in how this is implemented. Since the POLL step is the basis of the convergence analysis, it is the part of the algorithm where most research has been concentrated.

Lewis and Torczon [39] recognized that POLL should consider points on the mesh neighboring the incumbent solution in a set of directions whose nonnegative linear combinations span the space. This may seem simple, but it is a crucial observation. Coope and Price [21, 22, 23] extended this notion to the idea of frames, which can be thought of as doing away with the requirement that the POLL points be mesh neighbors. Audet and Dennis [13] suggested MADS as a way to implement frames so that the directions used in infinitely many POLL steps generate a dense set in the tangent cone at a MADS limit point $\hat{x} \in X$. This allows strong convergence results [13, 5] and excellent computational results for the MADS algorithms [15, 16, 41].

2.1 Some SEARCH strategies

The SEARCH step can be empty. By this we mean that the algorithm can be implemented as a sequence of POLL steps only. This is a reasonable choice when a local minimizer in the same basin as the initial guess is sufficient. Another reasonable strategy is to try a step in the same direction as a previously successful POLL step. It must be said that although

this seems reasonable, we understand that some researchers have found this approach of limited value at best.

We have experimented with random search as a SEARCH strategy. This has some success on the initial iteration, but it seems to be a waste of function values after that.

In our experience, the best SEARCH strategies involve the use of surrogates for f and C. We use *surrogate* to mean an inexpensive function that the user can employ to look extensively on the current mesh for points that the surrogate predicts will improve the current incumbent solution. Surrogates generally are of two types, simplified physics simulations and surfaces fit to a set of points in X usually chosen by some space filling design. We use the term *surrogate* rather than *approximation* because we do not want to imply that anything is required with respect to how well the surrogates approximate the problem functions [18].

Boeing uses DACE surrogates [46] in their Design Explorer filter implementation [10]. They generate data sites by an orthogonal array, and then fit a DACE model to the data. The SEARCH consists of a global Newton SQP method applied to the surrogate problem to try to generate several good local optimizers for that problem. Then they use the expensive "true" problem functions at those points to decide whether the SEARCH has been successful. Whenever new values of the true problem functions have been computed, they are used to recalibrate the surrogates. This surrogate management framework leads to very successful methods. Details are given in [10].

Alison Marsden has solved trailing edge shape design problems using both types of surrogates in an insightful way. She generates trial points using the MATLAB DACE surrogate package [40] and then uses a less expensive turbulence model to check whether a trial point is in X. If it is, then she runs the more accurate simulation. Her SEARCH consists of applying an evolutionary algorithm to the DACE surrogates. See [41] for details.

Another interesting application of surrogates is in [15], where a framework to identify good algorithmic parameter values is given. To illustrate this framework, MADS was applied to an objective function that measured the CPU time required by a trust-region algorithm [32] to solve a set of difficult problems. A natural surrogate function was constructed by having the trust-region method to solve a different list of easy problems.

2.2 The POLL step

The POLL step is more rigidly defined than the SEARCH step. It is necessarily called when the SEARCH fails to produce an unfiltered point. The POLL step consists of a local exploration around the current incumbent solution. The trial points are generated in some directions scaled by a mesh size parameter. When either the SEARCH or the POLL step is successful, then the mesh size parameter is either kept constant or increased. Otherwise, when both steps fail to generate an unfiltered point, the incumbent is declared to be a *mesh local optimizer* [21] and the mesh size parameter is decreased.

In GPS, the POLL directions were restricted to belong to some finite set. The GPS convergence results [12, 4] are closely tied to these fixed directions. Furthermore, there are some known examples [9] for which GPS falls short of converging to a satisfactory solution because of this restriction.

MADS overcomes this limitation by allowing a larger set of POLL directions. In fact, as k (the iteration counter) goes to infinity, the union of the normalized POLL directions over all k becomes dense in the unit sphere. This algorithmic construction allows stronger convergence results [13].

In some cases, incomplete derivative information may be available. For example, in some MDO problems, derivatives for some disciplines may be available, but not for others, and derivatives across multiple disciplines are not available. If the full gradient is available, directions can be chosen so that all but one are ascent directions, which can be ignored, thus reducing the required number of function evaluations to one per iteration [6]. In this case, MADS reduces to an approximation of steepest descent. Even if only some partial derivatives are known, MADS can exploit this information to reduce the number of function evaluations in each POLL step [6] without sacrificing theoretical convergence properties. In related work, Custódio and Vicente [26] compute a simplex gradient from a subset of previously evaluated points having certain geometrical properties, and they have

studied its use as a potential direction of descent in an effort to speed convergence.

Since MADS is opportunistic, in that it moves immediately to a new improved mesh point as soon as it is found, the order in which POLL points are evaluated can impact performance. One approach in which we have witnessed such improvement is what we call *dynamic polling*, in which the most recent successful direction is moved to the front of the queue after each successful POLL step. Dynamic polling was shown useful in [13] on a chemical engineering parameter fit problem [33]. If we were to use a surrogate in the SEARCH step, then evaluating the surrogate at each POLL point and then ordering them by surrogate function value would also be a wise choice. Custódio and Vicente [26] have also seen a reduction in function evaluations by computing a simplex gradient and ordering POLL points according to how small an angle the corresponding poll directions make with the negative of the simplex gradient. One must keep in mind, however, that these strategies (dynamic polling, surrogate and simplex gradient ordering) do not necessarily lead to improved computational times in all cases.

3. Why study these methods

In previous sections, we have mentioned some applications of MADS. In this section, we will make some general remarks about applications, but since the interested reader can find all the details we can furnish in the referenced papers, we save space here.

Also in this section we will discuss the theoretical support for MADS. We hope that other derivative-free optimization researchers will consider using nonsmooth analysis to analyze their methods. The discovery of MADS was a direct result of our nonsmooth analysis of GPS, and that has made us enthusiastic about building a bridge to this advanced theoretical part of our discipline.

3.1 Importance in practice

It is likely that every paper in these special issues will make a case for the practical importance of derivative-free optimization methods. We second everything the other authors say, but we will use our space here to try to make a couple of points that other authors may not make.

The first point is that, in spite of the formidable aspects of our target class of problems, we are often able to solve them quite efficiently. The main reason we were the first to solve them is that there are barriers to applying traditional derivative-based methods, and heuristic searches use too many function evaluations for these relatively expensive problem functions.

These problems typically take minutes to weeks for each function value, and many of them have what Tim Kelley [19], who also sees such problems, likes to call "hidden constraints". This second point means that one calls the simulation codes that must be run to evaluate the functions for perfectly innocuous arguments, and they fail. Furthermore, they fail after running for about the same length of time as when they succeed. In [18], this sort of failure happens to us roughly twice in every three function calls.

The main reason we have seen for these evaluation failures is that the function evaluations depend on runtime linking of single discipline solvers; e.g., separate structures and fluids codes. This is characteristic of multidisciplinary design optimization (MDO) problems [25, 8]. The interested reader will find a vast amount of MDO literature on the web.

Thus, to get the ancillary variables needed to evaluate the objective and constraints, one must do a multidisciplinary analysis, meaning the runtime linking of the codes. In our experience, an MDA can be thought of as solving a large system of nonlinear equations for which no Jacobian information is practical. In such a circumstance, there is little one can try except simple successive substitution or nonlinear Gauss-Seidel. This is sensitive to the order in which the blocks are processed, and it is apt to fail.

Another difficulty inherent to some of the target problems is that the functions are often contaminated with noise. It is not infrequent that evaluating a function twice at the same value of x returns slightly different values.

3.2 Theoretical support

These algorithms are intended to be applied to nonsmooth problems, or to any problems for which derivatives are impractical, even by finite differences. Typically, both the objective function and the con-

straints are evaluated by running a black box computer code. There is no way one can measure the smoothness of these functions.

The convergence results state that if the MADS algorithm is applied to such problems, then some optimality conditions are guaranteed. In [12] and [13] we give a hierarchy of convergence results based on various degrees of smoothness of the objective and constraints.

At the bottom of the hierarchy, we have a result that if the iterates produced by the algorithm are bounded, then there is an $\hat{x}$, which is the limit of mesh local optimizers on meshes that get infinitely fine. Assuming a bounded sequence of iterates is a standard assumption in nonlinear optimization, and it holds for our algorithms if the initial level set is bounded.

Then, by adding more smoothness, the local optimality results become successively stronger for a limit point $\hat{x}$. At the smoothest end of the hierarchy, we have that if f is strictly differentiable near $\hat{x}$, and if the constraint qualification that the tangent cone $T_\Omega(\hat{x})$ to the feasible region Ω at $\hat{x} \in \Omega$ is non-empty and full-dimensional, then the directional derivative f' satisfies

$$f'(\hat{x}; d) \geq 0 \quad \text{for every } d \in T_\Omega(\hat{x}).$$

This is the KKT first-order optimality condition: there are no feasible strict descent directions. In the unconstrained case, the tangent cone is the entire space, and this last condition becomes $\nabla f(\hat{x}) = 0$.

The intermediate results of the convergence analysis are based on different degrees of smoothness. The directional derivatives f' are not appropriate to deal with non-smooth functions, as they are undefined. We turned to the nonsmooth community and found exactly the analytical tool that we needed to analyze the convergence of our methods: the Clarke Calculus [20].

Clarke proposes a generalization $f^\circ(\hat{x}; d)$ of the directional derivative for locally Lipschitz functions, and generalizations [45, 20, 35] of the tangent cone; namely, the hypertangent cone $T_\Omega^H(\hat{x})$, the Clarke tangent cone $T_\Omega^{Cl}(\hat{x})$, and the contingent cone $T_\Omega^{Co}(\hat{x})$. Armed with these definitions, we can show that depending on the smoothness, the limit point $\hat{x}$ generated by MADS satisfies

$$f^\circ(\hat{x}; d) \ \geq \ 0$$

for every

$$d \ \in \ T_\Omega^H(\hat{x}), \ T_\Omega^{Cl}(\hat{x}) \ \text{ or in } \ T_\Omega^{Co}(\hat{x}).$$

Furthermore, in [5], we discover that with more smoothness (namely, that f is twice strictly differentiable near $\hat{x}$), $\hat{x}$ satisfies a second-order Clarke-KKT necessary condition for optimality that depends on a generalization of the Hessian matrix [34]. In fact, with additional assumptions, $\hat{x}$ satisfies a second-order Clarke-KKT sufficient condition for optimality, thus ensuring convergence of MADS to a local minimizer [5].

In stating these results, we make the assumption that the set of directions used infinitely often is dense in the hypertangent cone at $\hat{x}$. As stated earlier, MADS is designed specifically so that this can be accomplished, but in order to do it in practice, our selection of positive spanning directions is done randomly. Consequently, most of our convergence results are with probability one.

4. What is still needed

There are practical issues we still need to deal with for the class of problems discussed above. Anyone who has worked with users has had the experience of being told that the problem has a certain property, e.g., ten design variables, only to be told after solving the problem that it would be nice to be able to deal with one hundred design variables. This is a sure sign of progress in the project. In this section, we will give brief descriptions of some of the main issues raised by users after an initial success with the first formulation.

4.1 Categorical variables

Nonlinear mixed integer problems are hard enough, but many engineering design problems involve categorical variables. These are discrete variables constrained to a discrete set as a part of X. The problem functions cannot be evaluated unless all categorical variables take on feasible discrete values. For example, simulating an oil field with 25.3 oil wells is out of the question unless one interpolates and thereby increases the number of expensive simulations required.

We use the term *mixed variable programming* (MVP) to denote mathematical programming problems with both continuous and categorical variables [11]. An example is found in the design of a fixed-length thermal insulation system [36] in which the objective is to minimize the power required subject to some reasonable linear constraints.

In this problem, the system consists of a series of insulators of various material types and thicknesses, each pair of which is separated by a metal plate, called a *heat intercept*, to which power is applied to maintain it at a specified temperature. The material thicknesses and intercept temperatures are the continuous variables, while the number and types of insulators are categorical. In fact, the insulator types are not even numeric, although each material type can be mapped to the numeric value of its index into a list of seven possible material types that may be selected. A further interesting complication is that the number of insulators, which defines the problem dimension, is itself a design variable. This problem was solved numerically in [36] using the algorithm introduced in [11]. Realistic nonlinear constraints on system mass, tensile yield stress, and thermal contraction were added to the problem in [3], and the resulting problem was solved numerically using a pattern search filter method [7].

Because of the general lack of ordinality with categorical variables, MVP problems present some unique challenges. For example, there is no general notion of local optimality. To overcome this challenge, the user must provide a set-valued neighborhood function that defines the set of discrete neighbors at every point. Local optimality is then defined with respect to this function at the limit point. In the example above, given a design of the system, discrete neighbors were formed in 3 ways: swapping a single insulator for another of a different material, adding an insulator and heat intercept at any location (and adjusting the continuous variables appropriately), or deleting any insulator with its adjacent intercept. Once the algorithm is appropriately modified, we guarantee that the resulting solution could not be improved by moving to a neighboring point, as defined by these three classes of neighbors.

The main modification to the algorithm consists of augmenting the POLL step to include points in the set of discrete neighbors, along with other promising points [11]. Convergence properties of GPS for MVP problems with a smooth objective function and bound constraints on the continuous variables were established in [11] and extended to general linear constraints and nonsmooth functions in [2]. Convergence results for the GPS filter method for MVP problems with nonlinear constraints was introduced in [2, 7].

The class of MVP problems is actually quite common in practice, even though the field is very new, and there are some important algorithmic and structural considerations that merit further research.

4.2 Multiple objectives

It is almost always true that real optimization problems have multiple objectives. They may not appear in this form, but scratch the surface and they will. For example, a client might suggest a bound constraint on some function $\gamma(x)$. But, when asked about the value of the bound, the client will say it should be as small as possible. In other words, the constraint is really another objective.

Another way multiple objectives show themselves is in documenting the problem solution for the client. Presented with a solution to an optimization problem, the client (or his/her boss) will want to know how much better/worse the objective would be if a certain constraint were to be relaxed/tightened.

The reader will see in both cases the standard objective synthesis approach of minimizing a weighted sum of the individual objectives is not helpful. In both cases, the decision maker wants to trade off one objective against the others. What we need is to give the client a notion of the Pareto surface. To see a simple case of the deficiencies in the the weighted sum approach, see [29]. We do not recommend this approach; however, [28] is an interesting way to find a single important Pareto point.

Since the filter approach is based on multi-objective ideas, we hope that our filter approach can be adapted to provide helpful tools. However, this is not as straight forward as one might hope.

4.3 Ability to handle more decision variables

It would be useful to extend MADS to handle hundreds of decision variables, on problems where par-

allelism [37] alone would not be sufficient to solve the problem. As with all direct search methods, we expect to see the number of function values required to solve an arbitrary n dimensional problem increase much faster than n. Our goal is to find alternative direct search methods that slow the growth.

5. Conclusions

Direct search methods are here to stay as a valuable subarea of optimization. They are interesting theoretically, and they are indispensable in practice. These special issues will document many of the advances that have been made in the area, but much remains to be done.

We have sketched some useful properties and limitations of MADS algorithms. A researcher willing to build a strong theoretical background in nonsmooth analysis and learn to work with users will find this a satisfying and fruitful area in which to work. The experience of helping a user formulate and solve a problem thought to be intractable is the ultimate validation for an applied mathematician. Come on in, the water is fine.

A reader interested in obtaining software should visit [1, 24, 27, 38, 40, 42].

REFERENCES

[1] M. A. Abramson. *NOMADm optimization software*, `http://www.afit.edu/en/ENC/Faculty/MAbramson NOMADm.html`.

[2] M. A. Abramson, *Pattern Search Algorithms for Mixed Variable General Constrained Optimization Problems*, Ph.D. Thesis, Department of Computational and Applied Mathematics, Rice University, 2002.

[3] M. A. Abramson, *Mixed variable optimization of a load-bearing thermal insulation system using a filter pattern search algorithm*, Optim. Eng., 5 (2004), pp. 157–177.

[4] M. A. Abramson, *Second-order behavior of pattern search*, SIAM J. Optim., 16 (2005), pp. 515–530.

[5] M. A. Abramson and C. Audet, *Convergence of mesh adaptive direct search to second-order stationary points*, SIAM J. Optim., to appear.

[6] M. A. Abramson, C. Audet, and J. E. Dennis, Jr., *Generalized pattern searches with derivative information*, Math. Program., 100 (2004), pp. 3–25.

[7] M. A. Abramson, C. Audet, and J. E. Dennis, Jr., *On the convergence of filter pattern search algorithms for mixed variable constrained optimization problems*, Technical Report TR04-09, Department of Computational and Applied Mathematics, Rice University, 2004.

[8] N. M. Alexandrov and M. Y. Hussaini, editors, *Multidisciplinary Design Optimization: State of the Art*, SIAM, Philadelphia, 1997.

[9] C. Audet, *Convergence results for pattern search algorithms are tight*, Optim. Eng., 5 (2004), pp. 101–122.

[10] C. Audet, A. J. Booker, J. E. Dennis, Jr., P. D. Frank, and D. W. Moore, *A surrogate-model-based method for constrained optimization*, AIAA Paper 2000–4891, presented at the 8th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, California, 2000.

[11] C. Audet and J. E. Dennis, Jr., *Pattern search algorithms for mixed variable programming*, SIAM J. Optim., 11 (2000), pp. 573–594.

[12] C. Audet and J. E. Dennis, Jr., *Analysis of generalized pattern searches*, SIAM J. Optim., 13 (2003), pp. 889–903.

[13] C. Audet and J. E. Dennis, Jr., *Mesh adaptive direct search algorithms for constrained optimization*, SIAM J. Optim., to appear.

[14] C. Audet and J. E. Dennis, Jr., *A pattern search filter method for nonlinear programming without derivatives*, SIAM J. Optim., 14 (2004), pp. 980–1010.

[15] C. Audet and D. Orban, *Finding optimal algorithmic parameters using the mesh adaptive direct search algorithm*, Technical Report G-2004-96, Les Cahiers du GERAD, 2004.

[16] V. Béchard, C. Audet, and J. Chaouki, *Robust optimization of chemical processes using a MADS algorithm*, Technical Report G-2005-16, Les Cahiers du GERAD, 2005.

[17] A. J. Booker, A. R. Conn, J. E. Dennis, Jr., P. D. Frank, M. W. Trosset, and V. Torczon, *Global modeling for optimization: Boeing/IBM/Rice collaborative project 1995 final report*, Technical Report ISSTECH–95–032, Boeing Information & Support Services, Research and Technology, M/S 7L–68, Seattle, Washington 98124, 1995.

[18] A. J. Booker, J. E. Dennis, Jr., P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset, *A rigorous framework for optimization of expensive functions by surrogates*, Struct. Optim., 17 (1999), pp. 1–13.

[19] T. D. Choi, O. J. Eslinger, C. T. Kelley, J. W. David, and M. Etheridge, *Optimization of automotive valve train components with implicit filtering*, Optim. Eng., 1 (2000), pp. 9–28.

[20] F. H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley & Sons, New York, 1983. Reissued in 1990 by SIAM, Philadelphia, as Volume 5 in *Classics in Applied Mathematics*.

[21] I. D. Coope and C. J. Price, *Frame-based methods for unconstrained optimization*, J. Optim. Theory Appl., 107 (2000), pp. 261–274.

[22] I. D. Coope and C. J. Price, *On the convergence of grid-based methods for unconstrained optimization*, SIAM J. Optim., 11 (2001), pp. 859–869.

[23] I. D. Coope and C. J. Price, *Positive bases in numerical optimization*, Comput. Optim. Appl., 21 (2002), pp. 169–175.

[24] G. Couture, C. Audet, J. E. Dennis, Jr., and M. A. Abramson. *The NOMAD project*. `http://www.gerad.ca/NOMAD`.

[25] E. Cramer, J. E. Dennis, Jr., P. D. Frank, R. M. Lewis, and G. R. Shubin, *On the convergence of grid-based methods for unconstrained optimization*, SIAM J. Optim., 11 (2001), pp. 859–869.

[26] A. L. Custódio and L. N. Vicente, *Using sampling and simplex derivatives in pattern search methods*, Technical Report 04-35, Department of Mathematics, University of Coimbra, 2004.

[27] A. L. Custódio and L. N. Vicente, *SID-PSM: A pattern search method guided by simplex derivatives for use in derivative-free optimization*, `http://www.mat.uc.pt/sid-psm`, 2005.

[28] I. Das, *On characterizing the 'knee' of the Pareto curve based on normal-boundary intersection*, Struct. Optim., 18 (1999), pp. 107–115.

[29] I. Das and J. E Dennis, Jr., *A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization programs*, Struct. Optim., 14 (1997), pp. 63–69.

[30] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, 1983. Reissued in 1996 by SIAM, Philadelphia, as Volume 16 in *Classics in Applied Mathematics*.

[31] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Math. Program., 91 (2002), pp. 239–269.

[32] N. I. M. Gould, D. Orban, and Ph. L. Toint. GALAHAD—*a library of thread-safe FORTRAN 90 packages for large-scale nonlinear optimization*, ACM Trans. Math. Software, 29 (2003), pp. 353–372.

[33] R. E. Hayes, F. H. Bertrand, C. Audet, and S. T. Kolaczkowski, *Catalytic combustion kinetics: Using a direct search algorithm to evaluate kinetic parameters from light-off curves*, Canad. J. Chem. Engrg., 81 (2003), pp. 1192–1199.

[34] J.-B. Hiriart-Urruty, J.-J. Strodiot, and V. H. Nguyen, *Generalized Hessian matrix and second-order optimality conditions for problems with $C^{1,1}$ data*, Appl. Math. Optim., 11 (1984), pp. 43–56.

[35] J. Jahn, *Introduction to the Theory of Nonlinear Optimization*, Springer-Verlag, Berlin, 1994.

[36] M. Kokkolaras, C. Audet, and J. E. Dennis, Jr., *Mixed variable optimization of the number and composition of heat intercepts in a thermal insulation system*, Optim. Eng., 2 (2001), pp. 5–29.

[37] T. G. Kolda and V. Torczon, *On the convergence of asynchronous parallel pattern search*, SIAM J. Optim., 14 (2004), pp. 939–964.

[38] Sandia National Laboratories, Asynchronous parallel pattern search, `http://software.sandia.gov/-appspack`, 2005.

[39] R. M. Lewis and V. Torczon, *Rank ordering and positive bases in pattern search algorithms*, Technical Report 96–71, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, Virginia, 1996.

[40] S. N. Lophaven, H. B. Nielsen, and J. Søndergaard, *DACE: A MATLAB kriging toolbox version 2.0*, Technical Report IMM-TR-2002-12, Technical University of Denmark, 2002.

[41] A. L. Marsden, *Aerodynamic Noise Control by Optimal Shape Design*, Ph.D. Thesis, Stanford University, 2004.

[42] Mathworks, Inc., MATLAB genetic algorithm and direct search (GADS) toolbox. `http://www.mathworks.com/products/gads`, 2005.

[43] C. J. Price and I. D. Coope, *Frame based ray search algorithms in unconstrained optimization*, J. Optim. Theory Appl., 116 (2003), pp. 259–377.

[44] C. J. Price and I. D. Coope, *Frames and grids in unconstrained and linearly constrained optimization: A nonsmooth approach*, SIAM J. Optim., 14 (2003), pp. 415–438.

[45] R. T. Rockafellar, *Generalized directional derivatives and subgradients of nonconvex functions*, Canad. J. Math., 32 (1980), pp. 257–280.

[46] T. J. Santner, B. J. Williams, and W. I. Notz, *The Design and Analysis of Computer Experiments*, Springer-Verlag, Berlin, 2003.

[47] V. Torczon, *On the convergence of pattern search algorithms*, SIAM J. Optim., 7 (1997), pp. 1–25.

Using Problem Structure in Derivative-Free Optimization

Philippe L. Toint
Department of Mathematics,
University of Namur,
Namur, Belgium
(philippe.toint@fundp.ac.be).

1. Introduction

Derivative-free optimization is the branch of optimization where the minimizer of functions of several variables is sought without any use of the objective's derivatives. Although a number of problems may involve constraints, we will focus in this paper on the unconstrained case, i.e.,

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1}$$

where f is an objective function which maps $\mathbb{R}^n$ into $\mathbb{R}$ and is bounded below, and where we assume that the gradient of f (and, a fortiori, its Hessian) cannot be computed for any x.

The main motivation for studying algorithms for solving this problem is the remarkably high demand from practitioners for such tools. In most of the case known to the author, the calculation of the objective function value $f(x)$ is typically *very expensive*, and its derivatives are not available either because $f(x)$ results from some physical, chemical or econometrical measure, or, more commonly, because it is the result of a possibly very large computer simulation, for which the source code is effectively unavailable. The occurrence of problems of this nature appears to be surprisingly frequent in the industrial world. For instance, we have heard of cases where evaluating $f(x)$ requires the controlled growth of a particular crop or the meeting of an adhoc committee. As can be guessed from this last example, the value of $f(x)$ may in practice be contaminated by noise, but the numerical techniques for handling this latter characteristic are also outside the scope of our presentation.

Several classes of algorithms are known for derivative-free optimization. A first class is that of direct search techniques, which includes the well-known and widely used simplex reflection algorithm of [22] or its modern variants [11, 27], the old Hookes and Jeeves method [20] or the parallel direct search algorithm initiated by Dennis and

Torczon [15, 29]. These methods are based on a predefined geometric pattern or grid and use this tool to decide where the objective function should be evaluated. A recent and comprehensive survey of the development in this class is available in [21] and the associated bibliography is accessible at `http://www.cs.wm.edu/~va/research/wilbur.html`. See also the paper by Abramson, Audet and Dennis in this issue. The main advantage of methods in this class is that they do not require smoothness (or even continuity) of the objective function.

A second algorithmic class of interest is that of interpolation/approximation techniques pioneered by Winfield [30, 31] and by Powell in a series of papers starting with [23, 24]. In these methods, a low-order polynomial (linear or quadratic) model of the objective function is constructed and subsequently minimized, typically in a trust-region context (see [6, Chapter 9]). Because they build smooth models, they are appropriate when the objective function is known to be smooth. A common feature of the algorithms of this type is that they construct a basis of the space of suitable polynomials and then derive the model by building the particular linear combination of the basis polynomials that interpolate (or sometimes approximate) known values of $f(x)$. Powell favors a basis formed of Lagrange fundamental polynomials, while Conn *et al.* [7, 8, 10] use Newton fundamental polynomials instead. Both choices have their advantages and drawbacks, which we will not discuss here. The polynomial space of interest is typically defined as the span of a given set of monomials: full quadratic polynomials in the two variables x_1 and x_2 are for instance those spanned by 1, x_1, x_2, x_1x_2, x_1^2 and x_2^2.

While methods in these two classes have been studied and applied in practice, their use has remained essentially limited to problems involving only a very moderate number of variables: the solution of problems in more than 20 variables is indeed possible in both cases, but is typically very computationally intensive. In direct search techniques, this cost is caused by the severe growth in the number of grid or pattern points that are necessary to "fill" $\mathbb{R}^{20}$ or spaces of even higher dimensions. A similar difficulty arises in interpolation methods, where a total of $(n+1)(n+2)/2$ known function values are necessary to define a full quadratic model, but it is also compounded with the relatively high complexity in linear algebra due to repetitive minimization/maximization of such quadratic polynomials. One may therefore wonder if there is any hope for the derivative-free solution of problems in higher dimensions. It is the purpose of this paper to indicate that this hope may not be unfunded, at least for problems that exhibits some structure. Section 2 discusses the partially separable structure which will be central here, while Sections 3 and 4 indicate how it can be numerically exploited in direct search and interpolation methods, respectively. We then consider the special case of sparse problems in Section 5. Some conclusions are finally presented in Section 6.

2. Problem structure

Large-scale optimization problems often involve different parts, or blocks. Each block typically has its own (small) set of variables and other (small) sets of variables that link the block with other blocks, resulting, when all blocks are considered together, in a potentially very large minimization problem. One may think, for instance, of a collection of chemical reaction tanks, each with its temperature, pressure or stirring controls, and with its input raw material and output products. Other examples include discretized problems where a given node of the discretization may involve more than a single variable but is only connected to a few neighboring nodes, or PDE problems with domain decomposition, or multiple-shooting techniques for trajectory/orbit calculations, where each orbital arc only depends on a few descriptive or control variables, with the constraint that the different arcs connect well via a (small) set of common variables. Examples of this type just abound, especially when the size of the problem grows. It is interesting that their structure can very often be captured by the notion of *partial separability*, introduced by Andreas Griewank and the author in [17, 19].

A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be *partially separable*, if its value may be expressed, for every x, as

$$f(x) = \sum_{i=1}^{q} f_i(U_i x), \tag{2}$$

where each *element function* $f_i(x)$ is a function de-

pending on the *internal variables* $U_i x$ with U_i being a $n_i \times n$ matrix ($n_i \ll n$). Very often, the matrix U_i happens to be a row subset of the the identity matrix, and the internal variables are then just a (small) subset of the original problem variables.

Griewank and Toint show in [18] that *every twice-continuously differentiable function with a sparse Hessian is partially separable*, which gives a hint at why such functions are so ubiquitous. It is remarkable that, although this is only an existence result and specifies a particular decomposition (2) only indirectly, we are not aware of any practical problem having a sparse Hessian and whose partially separable decomposition is not explicitly known.

Partially separable functions (and their extension to group-partially separable ones) have been instrumental in the design of several numerical codes, including the LANCELOT package [5]. We now intend to show how they can be exploited in derivative-free optimization.

3. Direct search for partially separable objective functions

For our description of the structure use in direct search methods, we focus on the common case where the matrices U_i are subsets of the identity, which then implies that each element function f_i only depends on a small subset of the problem variables. As in [28], we assume that each f_i is available individually, along with a list of the problem variables on which f_i depends. Two problem variables are then said to interact if at least one element f_i depends on both. Sets of non-interacting variables are useful because the change in f from altering all variables in the set is just the sum of the changes in f arising from altering each variable in the set individually.

Exploiting partial separability is advantageous for direct search methods in two separate ways. It can both reduce the computational cost of each function value sought, and also provide function values at related points as a by-product. For example, consider a totally separable function

$$f(x) = \sum_{i=1}^{n} f_i(x_i)$$

where $x = (x_1, \ldots, x_n)^T$, and where the value of each element f_i is known at a point x. Assume, as is typical in pattern search methods, that we wish to explore f in the vicinity of x by calculating f at each point of the form $\{x \pm he_i\}_{i=1}^n$, where h is a positive constant and e_i is the i^{th} column of the identity matrix. Calculating f at each point requires the evaluation of one f_i, and so the total cost of the $2n$ function values at the points $\{x \pm he_i\}_{i=1}^n$ amounts to *two* function evaluations of the complete function f. However, any set of changes to f from steps of the form $\pm he_i$ are independent of one another provided no two such steps alter the same variable. Hence as a by-product we obtain the function values at all other points of the form $x + h\sum_{i=1}^n \eta_i e_i$ where each η_i is either -1, 0, or 1. Thus we obtain f at $3^n - 1$ new points for the total cost of *two* function evaluations.

When f is only partially separable, the gains are not as dramatic as in the totally separable case, but they still can be very substantial. For example, let f be of the form

$$\begin{aligned} f(x) \;=\; & f_1(x_n, x_1, x_2) + \\ & \sum_{i=2}^{n-1} f_i(x_{i-1}, x_i, x_{i+1}) + \\ & f_n(x_{n-1}, x_n, x_1). \end{aligned} \tag{3}$$

Then, given each $f_i(x)$, calculating f at $\{x \pm he_i\}_{i=1}^n$ costs the equivalent of six complete function evaluations. When n is divisible by 3, an inductive argument shows that the number of other function values obtained as by-products is $7^{(n/3)} - 2n - 1$.

A third example shows that the two advantages of partial separability are distinct. Let $f = f_3(x_1, x_2) + f_2(x_1, x_3) + f_1(x_2, x_3)$. Calculating f at all points of the form $x \pm he_i$, $i = 1, 2, 3$, costs four function evaluations, given that each $f_i(x)$, $i = 1, \ldots, 3$ is known in advance. No further function values are generated as by-products.

We may now exploit these features in constructing a pattern search method along the lines described in [11], where minimization is conducted on a sequence of successively finer nested grids (each grid is a subset of its predecessor), aligned with the coordinate directions. The m^{th} grid $\mathcal{G}^{(m)}$ is of the form

$$\mathcal{G}^{(m)} = \left\{ x^{(0)} + 2^{1-m} \sum_{i=1}^{n} \eta_i e_i \mid \eta_i \in \mathbb{Z} \right\},$$

where $x^{(0)}$ is the initial point and $\mathbb{Z}$ is the set of

signed integers. Loosely speaking, each grid is obtained by taking its predecessor and 'filling in' all points half-way between each pair of existing grid points.

For each grid, the algorithm generates a finite subsequence of iterates on the grid, with monotonically decreasing objective function values. This subsequence is terminated when no lower grid point can be found around the current iterate (the final iterate is then called a grid local minimizer) and the algorithm then proceeds onto the next grid in the sequence.

The key idea here is to exploit partial separability in constructing the minimizing subsequences. This is done as follows. Before the first iteration, the algorithm starts by grouping the problem variables into subsets indexed by $V_1, \ldots, V_r$ such that all variables whose index are V_p appear in exactly the same element functions f_i's. This means that all these variables are equivalent in terms of which other variables they interact with. (In example (3), $V_p = \{p\}$ for $p = 1, \ldots, r = n$.) Each V_p then determines a subspace

$$S_p = \text{span}\{e_j\}_{i \in V_p}.$$

Thus, identifying these subspaces with their generating variables, we have that some S_p interact and some (hopefully most) are *non-interacting*. (In example (3) again, the only interactions are between S_{p-1}, S_p and S_{p+1} and between S_1 and S_n.) We next build a positive basis B_p for each S_p, that is a set of vectors such that any $v \in S_p$ can be written as a positive linear combination of the vectors of B_p. For V_p, this basis can be chosen, for instance, as

$$B_p = \{e_j\}_{j \in V_p} \cup \{-\sum_{j \in V_p} e_j\},$$

in which case it is also minimal (see [12, 14] for further developments on positive basis). Minimization on a given grid $\mathcal{G}^{(m)}$ is then achieved by repeating the following steps.

1. For each $p \in \{1, \ldots, r\}$, we first calculate the objective function reduction

$$\Delta_p = \min_{v_j \in B_p} \sum_{i=1}^{q} \left[f_i(x^{(k)} + v_j) - f_i(x^{(k)}) \right],$$

and denote by s_p the argument of the minimum. Because of our definition of the subspaces S_p, only a small fraction of the elements $f_i(x^{(k)} + v_j)$ must be computed for every v_j.

2. We next choose the best objective function reductions from a set of non-interacting subspaces, i.e., we select an index set $I \subseteq \{1, \ldots, r\}$ such that

$$\Delta_I = \sum_{p \in I} \Delta_p$$

is minimal and the subspaces $\{S_p\}_{p \in I}$ are non-interacting.

3. If $\Delta_I < 0$, we finally define

$$x^{(k+1)} = x^{(k)} + \sum_{p \in I} s_p,$$

(note that $f(x^{(k+1)}) = f(x^{(k)}) + \Delta_I$), and increment k.

This is repeated until a grid local minimum is located ($\Delta_I \geq 0$). The procedure is then stopped if the desired accuracy is reached, or, if this not the case, the grid is refined and a new grid minimization started.

Of course, this description of the algorithm remains schematic, and we refer the interested reader to [28] for further details and algorithmic variants. In particular, this reference describes a technique for computing the index sets $V_1, \ldots, V_r$ from the partially separable structure of the objective function. We also note that it is not crucial to solve the combinatorial problem of Step 2 exactly: we may, for instance, use a simple greedy algorithm to identify a suitable index set I.

Does this work in practice? Table 1 shows the performance in terms of complete function evaluations of the method just described (under the heading "PS", for partial separability), compared to the same algorithm without the exploitation of problems structure (under the heading "no st."). The comparison is made on two well-known test problems from the CUTEr test set [16].

The conclusion is very clear: using the partially separable structure in direct search methods makes their application to relatively large-scale problems possible, while the unstructured approach rapidly reaches its limits in size. It is also interesting to note that the methodology adapts in an obvious way if derivatives of some but not all element functions

n	LMINSURF		BROYDN3D	
	PS	no st.	PS	no st.
9	215	501	343	1241
16	483	3724	334	3605
25	484	10557	364	8087
36	890	19796	379	16503
49	1002	—	363	24531
64	1149	—	362	—
81	1413	—	389	—
100	1634	—	362	—
121	2045	—	362	—
144	2120	—	392	—
169	2689	—	361	—
196	3233	—	361	—
5625	79511	—	535	—

Table 1: Number of function evaluations required to minimize the linear minimum surface (LMINSURF) and Broyden tridiagonal (BROYDN3D) functions in various dimensions (from [28]).

are available, or if one is ready, for some or all p, to compute Δ_p by minimizing in the complete subspace S_p instead of only considering the grid points defined by the positive basis B_p.

4. Interpolation methods for partially separable problems

We now turn to interpolation techniques and focus on trust-region methods. At each iteration of this type of methods, (typically quadratic) model of the objective function is constructed, which interpolates a set of known functions values. In other words, one builds the quadratic $m^{(k)}$ such that

$$m^{(k)}(y) = f(y) \quad \text{for all} \quad y \in Y^{(k)},$$

where $Y^{(k)}$ is the *interpolation set* at iteration k, a set of points at which values of the objective are known. As indicated above, the number of points in $Y^{(k)}$ must be equal to $(n+1)(n+2)/2$ for estimating a fully quadratic $m^{(k)}$. This model is then minimized within the trust-region, and the resulting trial step accepted or rejected, depending on whether or not the achieved decrease in f sufficiently matches the predicted decrease in the model. The size of the trust-region is then reduced if the match is poor, or possibly enlarged if it is satisfactory. This broad algorithmic outline of course hides a number of practical issues that are critical to good numerical performance. The most important is that the geometry of the sets $Y^{(k)}$ (i.e., the repartition the interpolation points y in $\mathbb{R}^n$) must satisfy a condition called *poisedness*. If we consider a two-dimensional problem, for instance, it is rather intuitive that the points in $Y^{(k)}$ should not all lie on a straight line. This poisedness condition can be formalized and tested [8, 9, 26] in various ways, but the key observation is that the same set of points cannot be used for ever as the algorithm proceeds, or even only modified to include the objective values at the new iterates. Figure 1 illustrates the potentially disastrous effect on the model of a bad geometry (bottom), compared to a situation at a previous iteration where the geometry is adequately poised (top).

As a consequence, it is necessary to improve the geometry of $Y^{(k)}$ at some iterations, typically by computing the objective function's value at new points chosen to ensure sufficient poisedness.

How can we adapt this technique if we now assume that we know a partially separable decomposition (2) of the objective? (We no longer assume that the matrices U_i are row subsets of the identity.) The answer, which is fully developed in [4], is based on a fairly obvious observation: *instead of building $m^{(k)}$, a model of f in the neighborhood of an iterate $x^{(k)}$, we may now build a collection of models $\{m_i^{(k)}\}_{i=1}^q$, where each $m_i^{(k)}$ now models f_i in the same neighborhood.* Estimating a structured quadratic model of the form

$$m^{(k)}(x^{(k)} + s) = \sum_{i=1}^{q} m_i^{(k)}(x^{(k)} + s)$$

then requires at most $(n_{\max} + 1)(n_{\max} + 2)/2$ complete function evaluations, where we have defined $n_{\max} = \max_{i=1,\ldots,q} n_i$. Since $n_{\max} \ll n$, this is typically order(s) of magnitude less that what is required for estimating an unstructured model. Moreover, $n_{\max}$ is often independent of n.

However, we now have to manage q interpolation sets $Y_i^{(k)}$ $(i = 1, \ldots, q)$ over the iterations, instead of a single one. We then have to consider two possible cases: either it is possible for the user to evaluate a single element function $f_i(x)$ independently of the others (as we have assumed above), or the objective function must always be evaluated as a whole (i.e.,

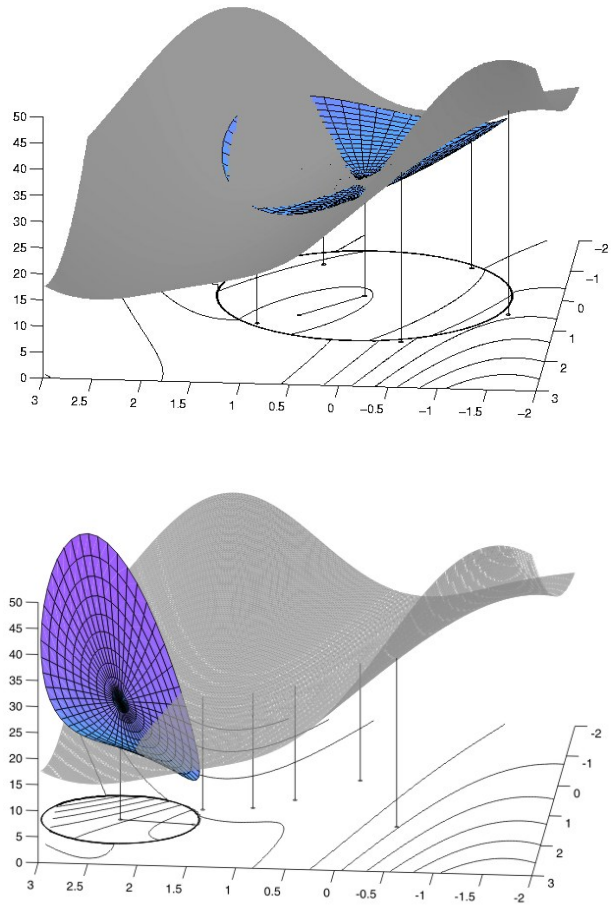

Figure 1: How the evolution of the interpolation set can lead to a bad model when its geometry deteriorates (the interpolation points are materialized by vertical bars from the surface to the plane, the model is quadratic and shown inside the trust-region)

only the collection of values $\{f_i(x)\}_{i=1}^q$ can be computed for a given x). In the second case, a straightforward implementation of the algorithm could be very expensive in terms of function evaluations if one blindly applies a geometry improvement procedure to each $Y_i^{(k)}$. Indeed, computing a vector that improves the geometry of the i-th interpolation set yields a vector having only n_i "useful" components, and it is not clear which values should be assigned to the remaining $n-n_i$ components. The strategy in this case is to group the necessary individual function evaluations by applying the CPR procedure of Curtis, Powell and Reid [13] for estimating sparse Jacobian matrices to the $n \times q$ occurrence matrix of variables x_j into elements f_i. (Other techniques involving graph coloring in the spirit of [1] are also possible). This typically results is substantial savings in terms of function evaluations.

As in the previous section, we do not elaborate more on the algorithmic details and variants (and refer to [4] for this), but rather illustrate the potential benefits of the approach on a few examples. These benefits are already visible for medium-size problems, as is shown in Table 2 where the unstructured algorithm (UDFO) is compared with its version using partially separable structure (PSDFO) on two CUTEr problems. The "I" version of PSDFO refers to the case where each element function f_i is accessible, and the "G" version to the case where the complete collection of $\{f_i(x)\}_{i=1}^q$ must be evaluated together for every x.

Problem	n	UDFO	PSDFO(I)	PSDFO(G)
ARWHEAD	10	311	30	54
	15	790	30	56
	20	1268	35	161
	25	1478	37	198
BDQRTIC	10	519	348	358
	15	1014	345	382
	20	1610	509	596
	25	2615	360	542

Table 2: Comparison of the number of function evaluations required by DFO solvers for solving medium-size instances of problems ARWHEAD and BDQRTIC (from [4]).

Again, the use of the partially separable structure brings clear benefits, and these are also more pronounced when the values of the elements can be computed individually (case "I"). Table 3 indicates that these benefits extend to higher dimensions, as expected.

5. Interpolation methods for sparse problems

We conclude our overview of the use of structure in derivative-free optimization by examining interpolation methods in the less favorable case where structure is present but access to the individual values of the element functions $f_i(x)$ impossible. This may

n	GENHUMPS		BROYDN3D	
	PSDFO(I)	PSDFO(G)	PSDFO(I)	PSDFO(G)
10	114	168	53	70
20	200	345	56	84
50	202	357	68	114
100	249	433	153	342
200	253	436	123	278

Table 3: The number of function evaluations required by PSDFO for solving problems GENHUMPS and BROYDN3D in various dimensions (from [4]).

happen, for instance, when the objective function value results from a complicated simulation involving a discretized partial differential equation (such as a complicated fluid calculation). In such cases, we often know that a partially separable decomposition of the form (2) exists (resulting, in our example, from the discretization topology), but we are only given the final value of $f(x)$, without its decomposition in its element function values (note the difference with the "G" case in the previous section). This implies that we typically know that the Hessian of f, $H(x)$, is sparse for every x, and know its structure. Can we exploit this (more limited) information?

As above, the idea is to construct a quadratic model $m^{(k)}$ that reflects the problem structure as much as possible: we thus need to construct *a model whose Hessian has the same sparsity structure as that of the objective function*. Interestingly, as noted in [2, 3], this is remarkably easy. Indeed, a given sparsity pattern is equivalent to a selection of a subset of the monomials generating the quadratic polynomials: if the (i, j)-th entry of $H(x)$ (and thus the (j, i)-th one) is known to be zero for all values of x, this simply indicates that f can be modelled by a restricted quadratic polynomial that does not involve the monomial $x_i x_j$. Thus the models $m^{(k)}$ may now be viewed as a linear combination of $1+n+n_H$ monomials, where n_H is the number of nonzeros in the lower triangular part of $H(x)$. This last number is often a small multiple of n, in which case the size of the interpolation set $Y^{(k)}$ is linear rather than quadratic in n. The cost of evaluating a sparse quadratic model is thus also very attractive, although typically larger than that of a partially separable model (often independent of n, as we have noted in the previous section).

These observations have been exploited in [2, 3], but also in [26] where Powell describes his successful UOBSQA code. The efficiency of this technique is attested by the results of Table 4, where the number of objective evaluations required for convergence is reported for UDFO (as in Table 2) and UOBSQA.

Problem	Dimension	UDFO	UOBSQA	UOBDQA
ARWHEAD	$n = 10$	311	118	105
	$n = 15$	790	170	164
	$n = 20$	1268	225	260
	$n = 25$	1478	296	277
BDQRTIC	$n = 10$	519	350	288
	$n = 15$	1014	594	385
	$n = 20$	1610	855	534
	$n = 25$	2615	1016	705

Table 4: Comparison of the number of function evaluations required by UDFO, UOBSQA and UOBDQA solvers for medium-size instances of problems ARWHEAD and BDQRTIC (from [4]).

Finally, one may argue that there is in fact no need that the sparsity structure of the model's Hessian really reflects that of $H(x)$. It is indeed possible to simply *impose an a priori sparsity structure* in order to reduce the size of the interpolation set, even if no sparsity information is available for $H(x)$. This idea was suggested in [2] and implemented, in its extreme form where the model's Hessian is assumed to be diagonal, in the UOBDQA code by Powell [25]. The excellent results obtained with this technique are also apparent in Table 4, even if one might guess that it will be mostly effective on problems whose Hessian is diagonally dominant. Potential extensions of this idea include the exploration of techniques (inspired for instance from automatic learning) which adaptively taylor the imposed sparsity pattern to the effective behaviour of the objective function.

6. Conclusions and perspectives

Given the discussion above, one may clearly conclude that *yes, the derivative-free optimization algorithms can exploit problem structure to* (sometimes dramatically) *improve their efficiency*. Moreover, the arguments presented also indicate that a richer structural information typically results in more substantial efficiency gains when using interpolation methods: the

number of objective function evaluations necessary to estimate a quadratic model indeed ranges from quadratic in n (no structure) via linear in n (when exploiting sparsity) to independent of n (when using partially separability).

It is clear that the development of structure-aware derivative-free optimization methods and packages is only starting, and much remains to be done. We think, in particular, of extensions of the ideas discussed here to the constrained case, and their applications to neighboring research areas, such as domain decomposition (an ongoing project explores the use of partial separability in this context) and others. Developments in these directions combine both the more abstract aspects of algorithm design and theory with the very practical nature of a subject in high industrial demand. There is no doubt that they therefore constitute valuable research challenges.

Acknowlegment

Thanks to Luis N. Vicente for suggesting the topic of this note, and to Annick Sartenaer, the queen of referees, whose comments improved its form.

REFERENCES

[1] T. F. Coleman and J. J. Moré, *Estimation of sparse Jacobian matrices and graph coloring problems*, SIAM J. Numer. Anal., 20 (1983), pp. 187–209.

[2] B. Colson and Ph. L. Toint, *Exploiting band structure in unconstrained optimization without derivatives*, Optim. Eng., 2 (2001), pp. 349–412.

[3] B. Colson and Ph. L. Toint, *A derivative-free algorithm for sparse unconstrained optimization problems*, in *Trends in Industrial and Applied Mathematics*, A. H. Siddiqi and M. Kočvara, editors, Kluwer Academic Publishers, Dordrecht, (2002) pp. 131–149.

[4] B. Colson and Ph. L. Toint, *Optimizing partially separable functions without derivatives*, Optim. Methods Softw., 20 (2005), pp. 493–508.

[5] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, LANCELOT*: A Fortran package for large-scale nonlinear optimization (Release A)*, in Springer Series in Computational Mathematics, Number 17, Springer-Verlag, Berlin, 1992.

[6] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust-Region Methods*, MPS-SIAM Series on Optimization, Number 1, SIAM, Philadelphia, 2000.

[7] A. R. Conn, K. Scheinberg, and Ph. L. Toint, *On the convergence of derivative-free methods for unconstrained optimization*, in *Approximation Theory and Optimization: Tributes to M. J. D. Powell*, A. Iserles and M. Buhmann, editors, Cambridge University Press, Cambridge, (1997) pp. 83–108.

[8] A. R. Conn, K. Scheinberg, and Ph. L. Toint, *Recent progress in unconstrained nonlinear optimization without derivatives*, Math. Program., 79 (1997), pp. 397–414.

[9] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Geometry of sample sets in derivative free optimization. Part I: polynomial interpolation*, Technical Report 03-09, Department of Mathematics, University of Coimbra, 2003. Revised September 2004.

[10] A. R. Conn and Ph. L. Toint, *An algorithm using quadratic interpolation for unconstrained derivative free optimization*, in *Nonlinear Optimization and Applications*, G. Di Pillo and F. Gianessi, editors, Plenum Publishing, New York, (1996) pp. 27–47. Also available as Report 95/6, Dept. of Mathematics, FUNDP, Namur, Belgium.

[11] I. D. Coope and C. J. Price, *On the convergence of grid-based methods for unconstrained optimization*, SIAM J. Optim., 11 (2001), pp. 859–869.

[12] I. D. Coope and C. J. Price, *Positive bases in numerical optimization*, Comput. Optim. Appl., 21 (2002), pp. 169–176.

[13] A. Curtis, M. J. D. Powell, and J. Reid, *On the estimation of sparse Jacobian matrices*, J. Inst. Math. Appl., 13 (1974), pp. 117–119.

[14] C. Davis, *Theory of positive linear dependence*, Amer. J. Math., 76 (1954), pp. 733–746.

[15] J. E. Dennis and V. Torczon, *Direct search methods on parallel machines*, SIAM J. Optim., 1 (1991), pp. 448–474.

[16] N. I. M. Gould, D. Orban, and Ph. L. Toint, CUTEr*, a constrained and unconstrained testing environment, revisited*, ACM Trans. Math. Software, 29 (2003), pp. 373–394.

[17] A. Griewank and Ph. L. Toint, *On the unconstrained optimization of partially separable functions*, in *Nonlinear Optimization 1981*, M. J. D. Powell, editor, Academic Press, London, (1982) pp. 301–312.

[18] A. Griewank and Ph. L. Toint, *Partitioned variable metric updates for large structured optimization problems*, Numer. Math., 39 (1982), pp. 119–137.

[19] A. Griewank and Ph. L. Toint, *On the existence of convex decomposition of partially separable functions*, Math. Program., 28 (1984), pp. 25–49.

[20] R. Hooke and T. A. Jeeves, *Direct search solution of numerical and statistical problems*, J. ACM, 8 (1961), pp. 212–229.

[21] T. G. Kolda, R. M. Lewis, and V. Torczon, *Optimization by direct search: New perspectives on some classical and modern methods*, SIAM Rev., 45 (2003), pp. 385–482.

[22] J. A. Nelder and R. Mead, *A simplex method for function minimization*, Comput. J., 7 (1965), pp. 308–313.

[23] M. J. D. Powell, *A direct search optimization method that models the objective and constraint functions by linear interpolation*, in *Advances in Optimization and Numerical Analysis, Proceedings of the Sixth Workshop on Optimization and Numerical Analysis, Oaxaca, Mexico*, S. Gomez and J. P. Hennart, editors, Kluwer Academic Publishers, Dordrecht, Volume 275 (1994), pp. 51–67.

[24] M. J. D. Powell, *A direct search optimization method that models the objective by quadratic interpolation*, Presentation at the 5th Stockholm Optimization Days, Stockholm, 1994.

[25] M. J. D. Powell, *On trust region methods for unconstrained minimization without derivatives*, Technical Report NA2002/02, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, 2002.

[26] M. J. D. Powell, *UOBYQA: Unconstrained optimization by quadratic interpolation*, Math. Program., 92 (2002), pp. 555–582.

[27] C. J. Price, I. D. Coope, and D. Byatt, *A convergent variant of the Nelder-Mead algorithm*, J. Optim. Theory Appl., 113 (2002), pp. 5–19.

[28] C. J. Price and Ph. L. Toint, *Exploiting problem structure in pattern-search methods for unconstrained optimization*, Optim. Methods Softw., 2006, to appear.

[29] V. Torczon, *On the convergence of the multidirectional search algorithm*, SIAM J. Optim., 1 (1991), pp. 123–145.

[30] D. Winfield, *Function and Functional Optimization by Interpolation in Data Tables*, Ph.D. Thesis, Harvard University, 1969.

[31] D. Winfield, *Function minimization by interpolation in a data table*, J. Inst. Math. Appl., 12 (1973), pp. 339–347.

Space Mapping for Engineering Optimization[1]

John W. Bandler
Simulation Optimization Systems Research Laboratory,
Department of Electrical and Computer Engineering,
McMaster University, Hamilton, ON, Canada L8S 4K1;
Bandler Corporation, Dundas, ON, Canada L9H 5E7
(bandler@mcmaster.ca).

Slawomir Koziel
Simulation Optimization Systems Research Laboratory,
Department of Electrical and Computer Engineering,
McMaster University, Hamilton, ON, Canada L8S 4K1
(koziels@mcmaster.ca).

Kaj Madsen
Informatics and Mathematical Modelling,
Technical University of Denmark,
DK-2800, Lyngby, Denmark
(km@imm.dtu.dk).

1. Introduction

Engineers have used optimization techniques for device and system modeling and design for decades [1]. Traditional techniques [2, 3] utilize simulations of appropriate models of the devices and any available derivatives to force relevant system responses to satisfy specifications subject to design constraints. The higher the fidelity (accuracy) of the models the more expensive we expect the application of traditional optimization to be. For complex problems this cost may be prohibitive.

Methodologies based on exploitation of iteratively refined surrogates of accurate or high-fidelity models address this issue. Through the construction of a suitably accurate physics-based surrogate model one can represent the objective function over a region of the design space. Then, instead of optimizing the high-fidelity model, one can optimize the surrogate which is further locally refined as increasingly accurate model data becomes available. Space mapping [4, 5, 6] is an example of this methodology. Such methods are called *surrogate-based* methods as opposed to the *direct* methods mentioned in the first paragraph.

There is a rich literature concerning surrogate-based optimization. Alexandrov *et al.* [7, 8, 9] de-

[1]This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grants OGP0007239 and STGP269760, and by Bandler Corporation.

scribe the so-called approximation and model management optimization technique. This assumes that the surrogate model satisfies so-called zero- and first-order consistency conditions with the high-fidelity model in question. Surrogate models based on second-order corrections are described in [10]. Dennis *et al.* [11, 12, 13] and Serafini [14] present a surrogate management framework and applications for engineering design. Surrogate optimization based on surface response approximation and kriging are discussed in [15, 16, 17]. Ong *et al.* [18] present evolutionary optimization via surrogate modeling. A survey and recommendations for the use of statistical approximation techniques in engineering design are given in [19]. Several review papers are available, including [20, 21, 22] and the recent paper [23].

We would like to emphasize that a characteristic feature that differentiates space mapping from several other surrogate-based optimization methods is that in our vision of space mapping, the surrogate model is constructed using an available, low-fidelity (and physically meaningful) model of the object response (the model being a function of the design variables), rather than pure interpolation/approximation. This is in keeping with the engineering tradition of developing for design purposes meaningful (not necessarily complex, often very simple) models of components of the physical world. Indeed, highly complex engineering component and system designs have been built before high-fidelity validations were computationally feasible.

In space mapping (SM), the objective function to be optimized is constructed from the responses of a so-called "fine model." By responses, we mean a vector of function values that represents the model's behavior for a given set of design parameter values, and from which any required constraint and objective function values are directly obtained. In the SM technology, conceived by Bandler in 1993, it is also assumed that there is an alternative set of functions available, not as accurate as those provided by the fine model but much faster to evaluate. These functions are derived from a so-called "coarse model." When the coarse model incorporates the same physics as the fine model, it is expected to yield its accuracy over a wide region of the parameter space. For example, in the radio-frequency (RF) and microwave area of electrical engineering, full-wave electromagnetic (EM) simulators can serve as fine models. Another example of a fine model is a physical experiment. Low-fidelity EM simulations or empirical electrical equivalent-circuit models could serve as coarse models. There is a vast library of such models in electrical engineering. Without such a library, the elements of which are continually being augmented and refined, electrical power systems, telecommunications circuits and systems, and computers would be literally unimaginable.

It was demonstrated in [4, 5], how SM can intelligently link coarse and fine models of different complexities in order to create a surrogate model that is almost as cheap to evaluate as the coarse model and (locally) almost as accurate as the fine model. In some engineering cases, the "coarse" model that is selected can even exhibit ideal or idealized behavior. The SM approach, either upfront or on the fly, updates the surrogate to better approximate the corresponding fine model in a region of interest.

In the first-proposed or original algorithm of Bandler *et al.* [4] the so-called coarse model is viewed as an idealization of the engineering device under consideration. As a result its optimal response is taken as the target response, i.e., the desired value of the objective function. The mapping between the parameter spaces of the coarse and fine models is called the space mapping. It maps available data points in the two spaces (i.e., fine and coarse model domains) which provide similar responses. It is evaluated in a process called parameter extraction (PE). In [4] surrogates are built based on linear approximations of the space mapping. Hence, in each iteration, the surrogate is a linearly mapped coarse model. The next iteration point is found as an optimal solution of the current surrogate.

A number of space mapping algorithms have been developed during last ten years, including aggressive space mapping (ASM) [5], trust-region ASM [24], implicit SM [25, 26], and output SM [27, 28, 29]. A review and exposition of advances in SM technology is contained in paper [6]. As we show in this paper, all of the existing space mapping approaches can be viewed as particular cases of one, generic formulation of space mapping.

Bandler *et al.* [6] offers a mathematical motivation, places SM into the context of classical optimization based on local Taylor approximations and

provides an extensive review of successful applications in many branches of engineering.

SM technology is recognized as a contribution to engineering design [30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40], especially in the microwave and RF arena. Snel (Philips, The Netherlands) used SM for new library models of RF components [32]. Hong and Lancaster [30] describe the aggressive SM algorithm as an elegant approach to microstrip filter design. Bakr *et al.* [33] employ artificial neural networks and Bandler *et al.* [34] study SM-based model enhancement. Ismail *et al.* [31] (Com Dev, Canada) use SM for the large-scale design of microwave filters and multiplexers for satellite communications. Pedersen *et al.* [35] utilize an output SM-based surrogate for modeling the thermo active components in new buildings. Ros *et al.* [36] use aggressive SM to design inductively coupled rectangular waveguide filters. Rautio [37] uses implicit SM for design of thick, tightly coupled conductors. He validates his model with a spiral inductor on silicon. Encica *et al.* [38] utilize SM to solve a shape optimization problem using Ansoft Maxwell2D. In automobile crashworthiness finite element simulations, each evaluation is expensive. Redhe and Nilsson [40] report that SM reduces the total computing time to optimize a vehicle's structure up to 50% compared with traditional optimization. SM has been applied to the complete finite element model of the new Saab 9-3 Sport Sedan. Intrusion into the passenger compartment area after the impact was reduced by 32% with no reduction in other crashworthiness responses.

Mathematicians are addressing mathematical interpretations of the formulation and convergence issues of SM algorithms [41, 42, 43, 44], although to date, convergence studies concerning SM consider only hybrid algorithms. In these papers, the authors utilized the general methodology of trust regions, made possible by their formulation of the response vector as a convex combination of the mapped coarse model and fine model response vectors. However, the convergence theories heavily rely on the combination with a classical Taylor-based method as a safeguard in the iteration. Therefore, classical principles of convergence proof are feasible. Unfortunately, it is not possible to prove convergence of "genuine" or pure SM algorithms in this way or explain their observed successful practical behavior because we don't necessarily have local model interpolation at the current iterate. Furthermore, tentative iterates may be accepted regardless of the improvement of the objective function (the fine model).

The development of the convergence theory for genuine SM algorithms is currently work in progress. In general, the conditions under which we can guarantee convergence of this class of algorithms concern SM and the engineering optimization problem itself (i.e., the accuracy of the coarse model as an approximation to the fine model). It follows that convergence depends on the quality of the match between the coarse and fine models. The convergence rate is also subject to the same consideration.

2. Optimization Using Surrogate Models

Let us state an engineering design problem as follows. Let $\boldsymbol{R}_f : X_f \to \mathbb{R}^m$ denote the response vector of a fine model of the engineering device, where $X_f \subseteq \mathbb{R}^n$. The vector $\boldsymbol{R}_f$ expresses the performance of the device, typically in terms of a measured output signal. In other words, we refer to "response" as a vector of function values associated with a given device. Our goal is to solve the problem

$$\boldsymbol{x}_f^* = \arg\min_{\boldsymbol{x}\in X_f} U(\boldsymbol{R}_f(\boldsymbol{x})) \tag{1}$$

where $U : \mathbb{R}^m \to \mathbb{R}$ is a given objective function. Note that the mathematical community typically refers to $U \circ \boldsymbol{R}_f$ as the objective function. We shall denote by X_f^* the set of solutions to (1) and call it the set of fine model minimizers.

We consider the fine model to be expensive to compute and solving (1) by direct optimization to be impractical. Instead, we use surrogate models, i.e., models that are not as accurate as the fine model but are computationally cheap, hence suitable for iterative optimization. We consider a general optimization algorithm that generates a sequence of points $\boldsymbol{x}^{(i)} \in X_f, i = 1, 2, \ldots$, and a family of surrogate models $\boldsymbol{R}_s^{(i)} : X_s^{(i)} \to \mathbb{R}^m, i = 0, 1, \ldots$, so that

$$\boldsymbol{x}^{(i+1)} = \arg\min_{\boldsymbol{x}\in X_f \cap X_s^{(i)}} U(\boldsymbol{R}_s^{(i)}(\boldsymbol{x})) \tag{2}$$

and $\boldsymbol{R}_s^{(i+1)}$ is constructed using suitable matching conditions with the fine model at $\boldsymbol{x}^{(i+1)}$ (and, per-

haps, some of the $\boldsymbol{x}^{(k)}, k = 1,\ldots,i$). If the solution to (2) is non-unique we may impose regularization. We may match responses, i.e.,

$$\boldsymbol{R}_s^{(i)}(\boldsymbol{x}^{(i)}) = \boldsymbol{R}_f(\boldsymbol{x}^{(i)}) \tag{3}$$

and/or match first-order derivatives

$$\boldsymbol{J}_{\boldsymbol{R}_s^{(i)}}(\boldsymbol{x}^{(i)}) = \boldsymbol{J}_{\boldsymbol{R}_f}(\boldsymbol{x}^{(i)}) \tag{4}$$

where $\boldsymbol{J}_{\boldsymbol{R}_s^{(i)}}$ and $\boldsymbol{J}_{\boldsymbol{R}_f}$ denote Jacobians of the surrogate and fine models, respectively. More precisely, we try to define models so that conditions such as (3) and (4) are satisfied.

3. SM-Based Surrogate Models

The family of surrogate models $\{\boldsymbol{R}_s^{(i)}\}$ can be implemented in various ways. SM assumes the existence of a so-called coarse model that describes the same object as the fine model: less accurate but much faster to evaluate. It takes advantage of this fact by shifting the optimization burden into the coarse model.

Let $\boldsymbol{R}_c : X_c \to \mathbb{R}^m$ denote the response vectors of the coarse model, where $X_c \subseteq \mathbb{R}^n$. By $\boldsymbol{x}_c^*$ we denote the optimal solution of the coarse model, i.e.,

$$\boldsymbol{x}_c^* = \arg\min_{\boldsymbol{x}\in X_c} U(\boldsymbol{R}_c(\boldsymbol{x})). \tag{5}$$

We denote by X_c^* the set of all $\boldsymbol{x} \in X_c$ satisfying (5) and call it the set of coarse model minimizers. In the SM framework, the family of surrogate models is constructed from the coarse model in such a way that each $\boldsymbol{R}_s^{(i)}$ is a suitable distortion of $\boldsymbol{R}_c$, such that given matching conditions are satisfied. In what follows, we discuss surrogate models that follow from original space mapping, input space mapping, output space mapping (OSM) and implicit space mapping (ISM).

3.1 The Original SM-Based Surrogate Model

The original SM assumes the existence of a mapping $\boldsymbol{P} : X_f \to X_c$ such that $\boldsymbol{R}_c(\boldsymbol{P}(\boldsymbol{x}_f)) \approx \boldsymbol{R}_f(\boldsymbol{x}_f)$ (proximity of $\boldsymbol{R}_c$ and $\boldsymbol{R}_f$ is measured using a suitable metric) on X_f or at least on some subset of X_f which is of our interest. For any given $\boldsymbol{x}_f \in X_f$, $\boldsymbol{P}(\boldsymbol{x}_f)$ is defined using parameter extraction

$$\boldsymbol{P}(\boldsymbol{x}_f) = \arg\min_{\boldsymbol{x}\in X_c} \|\boldsymbol{R}_c(\boldsymbol{x}) - \boldsymbol{R}_f(\boldsymbol{x}_f)\|. \tag{6}$$

This is illustrated in Figure 3.1.

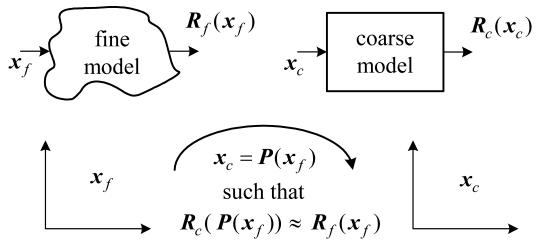

Figure 3.1 Space mapping $\boldsymbol{P}$.

In practical implementation, one may need to use regularization in order to assure existence of the space mapping $\boldsymbol{P}$ (i.e., existence and uniqueness of solution to (6) for any $\boldsymbol{x}_f$). This issue will not be dealt with in the present paper.

The surrogate model $\boldsymbol{R}_s^{(i)}$ is defined as

$$\boldsymbol{R}_s^{(i)}(\boldsymbol{x}) = \boldsymbol{R}_c(\boldsymbol{P}(\boldsymbol{x}^{(i)}) + \boldsymbol{B}^{(i)} \cdot (\boldsymbol{x} - \boldsymbol{x}^{(i)})) \tag{7}$$

for $i = 0, 1, \ldots,$ where $\boldsymbol{P}$ is defined by (6) and $\boldsymbol{B}^{(i)}$ is an approximation of $\boldsymbol{J}_{\boldsymbol{P}}(\boldsymbol{x}^{(i)})$, the Jacobian of $\boldsymbol{P}$ at $\boldsymbol{x}^{(i)}$, obtained using, e.g., the Broyden formula.

In a practical implementation, e.g., [5], instead of using directly the generic algorithm (2), the next iteration point $\boldsymbol{x}^{(i+1)}$ is obtained as a solution to the equation

$$\boldsymbol{P}(\boldsymbol{x}^{(i)}) + \boldsymbol{B}^{(i)}(\boldsymbol{x}^{(i)}) \cdot (\boldsymbol{x}^{(i+1)} - \boldsymbol{x}^{(i)}) = \bar{\boldsymbol{x}}_c^* \tag{8}$$

where $\bar{\boldsymbol{x}}_c^*$ is an element of X_c^* fixed for later reference (this formulation allows us to overcome the problem of non-uniqueness of the solution to optimization problem (5)).

3.2 The Input SM-Based Surrogate Model

The input SM aims at reducing misalignment between the fine and coarse models using an affine variable transformation established based on the available fine model data. The surrogate model $\boldsymbol{R}_s^{(i)}$ is

defined as

$$\boldsymbol{R}_s^{(i)}(\boldsymbol{x}) = \boldsymbol{R}_c(\boldsymbol{B}^{(i)} \cdot \boldsymbol{x} + \boldsymbol{c}^{(i)}) \tag{9}$$

$$(\boldsymbol{B}^{(i)}, \boldsymbol{c}^{(i)}) = \arg\min_{(\boldsymbol{B},\ \boldsymbol{c})} \varepsilon^{(i)}(\boldsymbol{B}, \boldsymbol{c}) \tag{10}$$

where matrices $\boldsymbol{B}^{(i)} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{c}^{(i)} \in \mathbb{R}$ are obtained using parameter extraction applied to the matching condition $\varepsilon^{(i)}$. Matching condition $\varepsilon^{(i)}$ determines the surrogate model as much as formula (9) does. We can consider different matching conditions that aim to match the fine and surrogate model responses and/or their first-order derivatives. A general form of the matching condition is

$$\varepsilon^{(i)}(\boldsymbol{B}, \boldsymbol{c}) = \sum_{k=0}^{i} w_k \|\boldsymbol{R}_f(\boldsymbol{x}^{(k)}) - \boldsymbol{R}_c(\boldsymbol{B} \cdot \boldsymbol{x}^{(k)} + \boldsymbol{c})\| + \sum_{k=0}^{i} v_k \|\boldsymbol{J}_{\boldsymbol{R}_f}(\boldsymbol{x}^{(k)}) - \boldsymbol{J}_{\boldsymbol{R}_c}(\boldsymbol{B} \cdot \boldsymbol{x}^{(k)} + \boldsymbol{c}) \cdot \boldsymbol{B}\|. \tag{11}$$

We assume that coefficients w_k and v_k are either 0 or 1 (although more general situations are conceivable in practice). Setting $w_k = 1, k = 0, \ldots, i$ and $v_k = 0, k = 0, \ldots, i-1, v_i = 1$ means that the surrogate tries to match the fine model response at all previous points $\boldsymbol{x}^{(k)}$ (including the current point) as well as the Jacobian at the current point.

3.3 The Output SM-Based Surrogate Model

The output space mapping (OSM) aims at reducing misalignment between the coarse and fine models by adding a difference (residual) between those two to $\boldsymbol{R}_c$. We define the function $\Delta\boldsymbol{R} : X_f \cap X_c \to \mathbb{R}^m$ as

$$\Delta\boldsymbol{R}(\boldsymbol{x}) = \boldsymbol{R}_f(\boldsymbol{x}) - \boldsymbol{R}_c(\boldsymbol{x}). \tag{12}$$

We construct surrogates that use (local) models of $\Delta\boldsymbol{R}$, denoted as $\Delta\boldsymbol{R}_m$. A generic surrogate model defined by OSM is

$$\boldsymbol{R}_s^{(i)}(\boldsymbol{x}) = \boldsymbol{R}_c(\boldsymbol{x}) + \Delta\boldsymbol{R}_m(\boldsymbol{x}, \boldsymbol{x}^{(i)}). \tag{13}$$

We consider the zero-order model $\Delta\boldsymbol{R}_m(\boldsymbol{x}, \boldsymbol{x}^{(i)}) = \Delta\boldsymbol{R}(\boldsymbol{x}^{(i)})$ which leads to the surrogate

$$\boldsymbol{R}_s^{(i)}(\boldsymbol{x}) = \boldsymbol{R}_c(\boldsymbol{x}) + \Delta\boldsymbol{R}(\boldsymbol{x}^{(i)}). \tag{14}$$

The second model is a first-order approximation of $\Delta\boldsymbol{R}$ of the form $\Delta\boldsymbol{R}_m(\boldsymbol{x}, \boldsymbol{x}^{(i)}) = \Delta\boldsymbol{R}(\boldsymbol{x}^{(i)}) + \boldsymbol{J}_{\Delta\boldsymbol{R}}(\boldsymbol{x}^{(i)}) \cdot (\boldsymbol{x} - \boldsymbol{x}^{(i)})$, where $\boldsymbol{J}_{\Delta\boldsymbol{R}}(\boldsymbol{x}^{(i)})$ denotes the Jacobian of $\Delta\boldsymbol{R}$ at $\boldsymbol{x}^{(i)}$. This leads to the surrogate

$$\boldsymbol{R}_s^{(i)}(\boldsymbol{x}) = \boldsymbol{R}_c(\boldsymbol{x}) + \Delta\boldsymbol{R}(\boldsymbol{x}^{(i)}) + \boldsymbol{J}_{\Delta\boldsymbol{R}}(\boldsymbol{x}^{(i)}) \cdot (\boldsymbol{x} - \boldsymbol{x}^{(i)}). \tag{15}$$

Instead of the exact Jacobian (usually unavailable) we can use its approximation produced by the Broyden update.

3.4 The Implicit SM-Based Surrogate Model

Implicit space mapping (ISM) makes use of additional parameters available in the coarse model, i.e., we have $\boldsymbol{R}_c : X_c \times X_p \to \mathbb{R}^m$ where $X_p \subseteq \mathbb{R}^q$ is the domain of such preassigned parameters. Preassigned (non-optimized) parameters abound in engineering design. Their successful exploitation as surrogate modeling parameters depends on much the same engineering expertise required in designating the optimization variables themselves.

An ISM optimization algorithm aims at predistortion of the coarse model by adjustment of certain preassigned parameters $\boldsymbol{x}_p$ so that, at the current point $\boldsymbol{x}^{(i)}$, the fine and coarse model response vectors are aligned. The predistorted model becomes a surrogate which, in turn, is optimized in order to obtain the next point $\boldsymbol{x}^{(i+1)}$. Thus, the surrogate model defined by ISM is

$$\boldsymbol{R}_s^{(i)}(\boldsymbol{x}) = \boldsymbol{R}_c(\boldsymbol{x}, \boldsymbol{x}_p^{(i)}) \tag{16}$$

where $\boldsymbol{x}_p^{(i)}$ is determined by solving a PE problem of the form

$$\boldsymbol{x}_p^{(i)} = \arg\min_{\boldsymbol{x}\in X_p} \|\boldsymbol{R}_f(\boldsymbol{x}^{(i)}) - \boldsymbol{R}_c(\boldsymbol{x}^{(i)}, \boldsymbol{x})\|. \tag{17}$$

3.5 SM Surrogate Models Based on Combined Concepts

It is possible and utilized in practice to combine the concepts discussed so far. For example, one can define the ith surrogate $\boldsymbol{R}_s^{(i)}$ using input, output and implicit SM as follows

$$\boldsymbol{R}_s^{(i)}(\boldsymbol{x}) = \boldsymbol{R}_c(\boldsymbol{B}^{(i)} \cdot \boldsymbol{x} + \boldsymbol{c}^{(i)}, \boldsymbol{x}_p^{(i)}) + \boldsymbol{d}^{(i)} + \boldsymbol{E}^{(i)} \cdot (\boldsymbol{x} - \boldsymbol{x}^{(i)}) \tag{18}$$

where matrices $\boldsymbol{B}^{(i)}$ and $\boldsymbol{c}^{(i)}$ as well as preassigned parameter values $\boldsymbol{x}_p^{(i)}$ are determined using parameter extraction (see (10), (11) and (17), respectively), while $\boldsymbol{d}^{(i)} = \boldsymbol{R}_f(\boldsymbol{x}^{(i)}) - \boldsymbol{R}_c(\boldsymbol{B}^{(i)} \cdot \boldsymbol{x}^{(i)} + \boldsymbol{c}^{(i)}, \boldsymbol{x}_p^{(i)})$, $\boldsymbol{E}^{(i)} = \boldsymbol{J}_{\boldsymbol{R}_f}(\boldsymbol{x}^{(i)}) - \boldsymbol{J}_{\boldsymbol{R}_c}(\boldsymbol{B}^{(i)} \cdot \boldsymbol{x}^{(i)} + \boldsymbol{c}^{(i)}, \boldsymbol{x}_p^{(i)}) \cdot \boldsymbol{B}^{(i)}$.

Combining different kinds of space mapping allows us to improve the flexibility of the surrogate model. On the other hand, the proper choice of the SM used to construct the surrogate, as well as the amount of fine model data used in this process, is usually problem dependent and knowledge of the problem and engineering experience are key factors to making this choice successful.

4. Conclusions

We have reviewed the space mapping approach to engineering surrogate modeling and design optimization. As with other surrogate methodologies, the aim is to avoid expensive direct optimization of high-fidelity models. In space mapping, we represent the objective function and constraint functions over a region of the design space through the construction of a suitably accurate physics-based surrogate. Instead of optimizing the high-fidelity model, we optimize the surrogate, which can further be refined as increasingly accurate model data becomes available. The notion of parameter extraction is important to space mapping. Here, high-fidelity data is exploited to validate the design and to improve the local alignment between the surrogate and the high-fidelity model. Using a low-fidelity and physically meaningful model to construct a surrogate is what differentiates space mapping from many other surrogate-based optimization methods. We have reviewed the original formulation as well as the so-called input, output and implicit formulations. Space mapping allows an engineer to exploit his/her detailed knowledge of the engineering design problem.

Matlab engines to implement the current state of the art (several dozen space mapping algorithms and models) to exploit full-wave electromagnetic simulators and fast, empirical, coarse or surrogate device models are under development. This endeavor is designed to make our technology universally available. The reader interested in this software should contact the first author.

REFERENCES

[1] M. B. Steer, J. W. Bandler, and C. M. Snowden, *Computer-aided design of RF and microwave circuits and systems,* IEEE Trans. Microwave Theory Tech., 50 (2002), pp. 996–1005.

[2] J. W. Bandler and S. H. Chen, *Circuit optimization: The state of the art,* IEEE Trans. Microwave Theory Tech., 36 (1988), pp. 424–443.

[3] J. W. Bandler, W. Kellermann, and K. Madsen, *A superlinearly convergent minimax algorithm for microwave circuit design,* IEEE Trans. Microwave Theory Tech., 33 (1985), pp. 1519–1530.

[4] J. W. Bandler, R. M. Biernacki, S. H. Chen, P. A. Grobelny, and R. H. Hemmers, *Space mapping technique for electromagnetic optimization,* IEEE Trans. Microwave Theory Tech., 42 (1994), pp. 2536–2544.

[5] J. W. Bandler, R. M. Biernacki, S. H. Chen, R. H. Hemmers, and K. Madsen, *Electromagnetic optimization exploiting aggressive space mapping,* IEEE Trans. Microwave Theory Tech., 43 (1995), pp. 2874–2882.

[6] J. W. Bandler, Q. S. Cheng, S. A. Dakroury, A. S. Mohamed, M. H. Bakr, K. Madsen, and J. Søndergaard, *Space mapping: The state of the art*, IEEE Trans. Microwave Theory Tech., 52 (2004), pp. 337–361.

[7] N. M. Alexandrov and R. M. Lewis, *An overview of first-order model management for engineering optimization,* Optim. Eng., 2 (2001), pp. 413–430.

[8] N. M. Alexandrov, R. M. Lewis, C. R. Gumbert, L. L. Green, and P. A. Newman, *Approximation and model management in aerodynamic optimization with variable-fidelity models,* AIAA Journal of Aircraft, 38 (2001), pp. 1093–1101.

[9] N. M. Alexandrov, J. E. Dennis, Jr., R. M. Lewis, and V. Torczon, *A trust region framework for managing use of approximation models in optimization,* Struct. Optim., 15 (1998), pp. 16–23.

[10] M. S. Eldred, A. A. Giunta, and S. S. Collis, *Second-order corrections for surrogate-based optimization with model hierarchies,* paper AIAA-2004-4457 in Proc. 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conf., Albany, NY, Aug. 30-Sept. 1, 2004.

[11] A. L. Marsden, M. Wang, J. E. Dennis, Jr. and P. Moin, *Optimal aeroacoustic shape design using the surrogate management framework*, Optim. Eng., 5 (2004), pp. 235–262.

[12] A. J. Booker, J. E. Dennis, Jr., P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset, *A rigorous framework for optimization of expensive functions by surrogates,* Struct. Optim., 17 (1999), pp. 1–13.

[13] J. E. Dennis, Jr. and V. Torczon, *Managing approximation models in optimization,* in *Multidisciplinary Design Optimization*, N. M. Alexandrov and M. Y. Hussaini, editors, SIAM, Philadelphia, (1997) pp. 330–374.

[14] D. B. Serafini, *A Framework for Managing Models in Nonlinear Optimization of Computationally Expensive Functions*, Ph.D. Thesis, Department of Computational and Applied Mathematics, Rice University, 2002.

[15] S. J. Leary, A. Bhaskar, and A. J. Keane, *A knowledge-based approach to response surface modeling in multifidelity optimization,* J. Global Optim., 26 (2003), pp. 297–319.

[16] S. J. Leary, A. Bhaskar, and A. J. Keane, *A derivative based surrogate model for approximating and optimizing the output of an expensive computer simulation*, J. Global Optim., 30 (2004), pp. 39–58.

[17] S. E. Gano, J. E. Renaud, and B. Sanders, *Variable fidelity optimization using a kriging based scaling function,* Proc. 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conf., Albany, New York, 2004.

[18] Y. S. Ong, P. B. Nair, and A. J. Keane, *Evolutionary optimization of computationally expensive problems via surrogate modeling,* American Institute of Aeronautics and Astronautics Journal, 41 (2003), pp. 687–696.

[19] T. W. Simpson, J. Peplinski, P. N. Koch, and J. K. Allen, *Metamodels for computer-based engineering design: survey and recommendations,* Engineering with Computers, 17 (2001), pp. 129–150.

[20] J.-F. M. Barthelemy and R. T. Haftka, *Approximation concepts for optimum structural design — a review,* Struct. Optim., 5 (1993), pp. 129–144.

[21] T. W. Simpson, A. J. Booker, D. Ghosh, A. A. Giunta, P. N. Koch, and R.-J. Yang, *Approximation methods in multidisciplinary analysis and optimization: A panel discussion,* 3rd ISSMO/AIAA Internet Conf. on Approximations in Optimization, Oct. 14-18, 2002.

[22] V. Torczon and M. W. Trosset, *Using approximations to accelerate engineering design optimization,* Proc. 7th AIAA/USAF/NASA/ISSMO Symp. Multidisciplinary Analysis and Optimization, St. Louis, Sept. 2-4, 1998.

[23] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidynathan, and P. K. Tucker, *Surrogate-based analysis and optimization,* Progress in Aerospace Sciences, 41 (2005), pp. 1–28.

[24] M. H. Bakr, J. W. Bandler, R. M. Biernacki, S. H. Chen, and K. Madsen, *A trust region aggressive space mapping algorithm for EM optimization,* IEEE Trans. Microwave Theory Tech., 46 (1998), pp. 2412–2425.

[25] J. W. Bandler, Q. S. Cheng, D. M. Hailu, and N. K. Nikolova, *A space mapping design framework,* IEEE Trans. Microwave Theory Tech., 52 (2004), pp. 2601–2610.

[26] J. W. Bandler, Q. S. Cheng, N. K. Nikolova, and M. A. Ismail, *Implicit space mapping optimization exploiting preassigned parameters,* IEEE Trans. Microwave Theory Tech., 52 (2004), pp. 378–385.

[27] J. W. Bandler, D. M. Hailu, K. Madsen, and F. Pedersen, *A space-mapping interpolating surrogate algorithm for highly optimized EM-based design of microwave devices,* IEEE Trans. Microwave Theory Tech., 52 (2004), pp. 2593–2600.

[28] J. W. Bandler, Q. S. Cheng, D. Gebre-Mariam, K. Madsen, F. Pedersen, and J. Søndergaard, *EM-based surrogate modeling and design exploiting implicit, frequency and output space mappings,* IEEE MTT-S Int. Microwave Symp. Dig., Philadelphia, (2003) pp. 1003–1006.

[29] S. Koziel, J. W. Bandler, and K. Madsen, *Towards a rigorous formulation of the space mapping technique for engineering design,* Proc. Int. Symp. Circuits Syst., ISCAS, 1 (2005), pp. 5605–5608.

[30] J.-S. Hong and M. J. Lancaster, *Microstrip Filters For RF/Microwave Applications*, John Wiley & Sons, New York, (2001) pp. 295–299.

[31] M. A. Ismail, K. G. Engel, and M. Yu, *Multiple space mapping for RF T-switch design,* 2004 IEEE MTT-S Int. Microwave Symp. Dig., Fort Worth, 3 (2004), pp. 1569–1572.

[32] J. Snel, *Space mapping models for RF components,* presented at the 2001 IEEE MTT-S Int. Microwave Symp. Workshop on Statistical Design and Modeling Techniques for Microwave CAD, Phoenix, 2001.

[33] M. H. Bakr, J. W. Bandler, M. A. Ismail, J. E. Rayas-Sánchez, and Q. J. Zhang, *Neural space mapping optimization for EM-based design,* IEEE Trans. Microwave Theory Tech., 48 (2000), pp. 2307–2315.

[34] J. W. Bandler, N. Georgieva, M. A. Ismail, J. E. Rayas-Sánchez, and Q. J. Zhang, *A generalized space mapping tableau approach to device modeling,* IEEE Trans. Microwave Theory Tech., 49 (2001), pp. 67–79.

[35] F. Pedersen, P. Weitzmann, and S. Svendsen, *Modeling thermally active building components using space mapping,* Proc. 7th Symp. Building Physics in the Nordic Countries, (2005) pp. 896–903.

[36] J. V. M. Ros, P. S. Pacheco, H. E. Gonzalez, V. E. B. Esbert, C. B. Martin, M. T. Calduch, S. C. Borras, and B. G. Martinez, *Fast automated design of waveguide filters using aggressive space mapping with a new segmentation strategy and a hybrid optimization algorithm,* IEEE Trans. Microwave Theory Tech., 53 (2005), pp. 1130–1142.

[37] J. C. Rautio, *A space-mapped model of thick, tightly coupled conductors for planar electromagnetic analysis,* IEEE Microwave Magazine, 5 (2004), pp. 62–72.

[38] L. Encica, D. Echeverria, E. Lomonova, A. Vandenput, P. Hemker, and D. Lahaye, *Efficient optimal design of electromagnetic actuators using space-mapping,* 6th World Congress on Structural and Multidisciplinary Optimization, Rio de Janeiro, Brazil, 2005.

[39] S. J. Leary, A. Bhaskar, and A. J. Keane, *A constraint mapping approach to the structural optimization of an expensive model using surrogates*, Optim. Eng., 2 (2001), pp. 385–398.

[40] M. Redhe and L. Nilsson, *Optimization of the new Saab 9-3 exposed to impact load using a space mapping technique*, Struct. Optim., 27 (2004), pp. 411–420.

[41] L. N. Vicente, *Space mapping: models, sensitivities, and trust-region methods*, Optim. Eng., 4 (2003), pp. 159–175.

[42] M. Hintermüller and L. N. Vicente, *Space mapping for optimal control of partial differential equations*, SIAM J. Optim., 15 (2005), pp. 1002–1025.

[43] M. H. Bakr, J. W. Bandler, K. Madsen, and J. Søndergaard, *An introduction to the space mapping technique*, Optim. Eng., 2 (2001), pp. 369–384.

[44] K. Madsen and J. Søndergaard, *Convergence of hybrid space mapping algorithms,* Optim. Eng., 5 (2004), pp. 145–156.

Bulletin

1. Event Announcements

HPOPT 2006 — 9th International Workshop on High Performance Optimization Techniques In honour of Kees Roos' 65th birthday

June 15–16, 2006

Delft University of Technology, The Netherlands

`http://stuwww.uvt.nl/~edeklerk/hpopt2006`

This is the 9th in a series of workshops on "high performance optimization techniques" that started in 1996 in Delft. It is also a special year for the organizers, as we celebrate the 65th birthday of Kees Roos as well as his career achievements. The workshop will consist of invited presentations by leading international experts in optimization, as well as a special session to celebrate Kees' birthday and career.

Confirmed invited speakers:

- Erling Andersen, Mosek
- Yanqin Bai, Shanghai Univ.
- Aharon Ben-Tal, Technion – Israel Inst. Technology
- Robert Freund, MIT
- Franois Glineur, Catholic Univ. Louvain
- Florian Jarre, Dusseldorf Univ.
- Monique Laurent, CWI, Amsterdam
- Yurii Nesterov, Catholic Univ. Louvain
- Franz Rendl, Univ. Klagenfurt
- Tams Terlaky, McMaster Univ.
- Takashi Tsuchiya, Inst. Statistical Mathematics, Tokyo
- Lieven Vandenberghe, UCLA
- Jean-Philippe Vial, Univ. Geneva

- Yinyu Ye, Stanford Univ.
- Akiko Yoshise, Univ. Tsukuba

19th International Symposium on Mathematical Programming
July 30 – August 4, 2006
Federal University of Rio de Janeiro, Brazil
`http://www.ismp2006.org`

The symposium will take place at the Federal University of Rio de Janeiro campus at Praia Vermelha, a charming district area conveniently located just a five-minute drive from some of the city's most famous attractions, such as the Sugar Loaf and Copacabana Beach. Shopping and restaurant facilities are also available within walking distance.

Founded in 1920, the Federal University of Rio de Janeiro – UFRJ is considered the oldest and largest federal university in the country as well as a model for the creation of several other public institutions of higher education. UFRJ has earned a solid international reputation as a center of excellence in teaching and research, with strategic importance to the development of the country. A highly qualified faculty, technical and administrative staff, modern labs, specialized libraries, undergraduate and graduate programs, numerous teaching, research and extension units, all have combined to build such reputation.

A particularly outstanding unit on Praia Vermelha campus is the University Palace, a beautiful nineteenth century neoclassical building. A former hospital for mentally ill people under Emperor D. Pedro II, the palace was only converted into the University President's Office in 1949. Following the transfer of the President's Office to Ilha do Fundo in the 60's, the University Palace was turned into a center for high level academic debate and artistic manifestation, a forum for science and culture.

The symposium will have five plenary sessions:

- Dantzig Memorial Session
- Gerard Cornuéjols, Carnegie Mellon Univ.
- Clovis Gonzaga, Federal Univ. Santa Catarina
- Arkadi Nemirovski, Technion – Israel Inst. Technology
- Alexander Shapiro, Georgia Inst. Technology

The list of semi-plenary speakers is the following:

- Daniel Bienstock, Columbia Univ.
- Maria Chudnovsky, Princeton Univ.
- Josef Hofbauer, Univ. College London
- Karl Kunisch, Univ. Graz
- Jean Lasserre, Laboratoire d'Analyse et d'Architecture des Systemes
- Jose Mario Martinez, State Univ. Campinas
- Yurii Nesterov, Catholic Univ. Louvain
- Satoru Iwata, Univ. Tokyo
- R. Ravi, Carnegie Mellon Univ.
- Toh Kim Chuan, National Univ. Singapore
- Robert Vanderbei, Princeton Univ.
- Vijay Vazirani, Georgia Inst. Technology

Papers on all theoretical, computational and practical aspects of mathematical programming are welcome. The presentation of very recent results is encouraged. The number of presentations is limited to one per speaker. However, no limit is imposed on the number of contributed/submitted abstracts per author, provided co-authors are the speakers. Only the speaker may submit an abstract. The deadline for abstract submission is June 10, 2006. More information is available at the web site. The e-mail contact is `ismp2006@cos.ufrj.br`.

2. Other Announcements

Applications of Algebraic Geometry at IMA

The Institute for Mathematics and its Applications (IMA) at the University of Minnesota announces membership opportunities in connection with its 2006-2007 thematic program on Applications of Algebraic Geometry.

Individuals may apply for four classes of membership at the IMA in connection with the 2006-2007 thematic program:

- IMA Postdoctoral Memberships
- IMA Industrial Postdoctoral Memberships
- IMA General Memberships
- IMA New Directions Visiting Professorships

IMA POSTDOCTORAL FELLOWSHIPS provide an excellent opportunity for mathematical scientists near the beginning of their career who have a background in and/or an interest in learning about applied and computational aspects of algebraic geometry. IMA postdoctoral fellowships run one to two years, at the option of the holder, starting September 5, 2006. In the second year of the appointment there are a variety of options to enhance career development, including participation in the 2007-2008 academic year program on Mathematics of Molecular and Cellular Biology, teaching, and working on an industrial project. Postdoctoral fellows receive a salary of $50,000 annually, and a travel allowance. Documentation of completion of all requirements for a doctoral degree in mathematics or a related area is required by the start of the appointment and within the last three years. (Deadline January 5, 2006.)

IMA INDUSTRIAL POSTDOCTORAL FELLOWSHIPS are designed to prepare mathematicians for research careers in industry or involving industrial interaction. IMA industrial postdoctoral fellowships run two years starting September 5, 2006. They are funded jointly by the IMA and an industrial sponsor, and holders devote 50own research and the IMA program and 50% effort working with industrial scientists. Industrial postdoctoral fellows receive a salary of $50,000 annually, and a travel allowance. Documentation of completion of all requirements for a doctoral degree in mathematics or a related area is required by the start of the appointment and within the last three years. (Deadline January 5, 2006.)

IMA GENERAL MEMBERSHIPS provide an opportunity for mathematicians and scientists employed elsewhere to spend a period of one month to one year in residence at the IMA, and to participate in the 2006-2007 thematic program. The residency should fall in the period September 2006 through June 2007 (in special cases extending into the summer months). Logistic support such as office space, computer facilities, and secretarial support will be provided, and local expenses may be provided. Preference will be given to supplementary support for persons with sabbatical leaves, fellowships, or other stipends. The research interests of General Members must relate to the thematic program and a doctoral degree is normally expected. Applications may be submitted at any time until the end of the thematic program, and will be considered as long as funds remain available. (Applications considered immediately and until funds are exhausted.)

IMA NEW DIRECTIONS VISITING PROFESSORSHIPS provide an extraordinary opportunity for established mathematicians (typically mid-career faculty at US universities) to branch into new directions and increase the impact of their research by spending the 2006-2007 academic year immersed in the thematic program at the IMA. Visiting Professors will enjoy an excellent research environment and stimulating scientific program connecting algebraic geometry and related areas of mathematics with a broad range of fields of application. New Directions Visiting Professors are expected to be resident and active participants in the program but are not assigned formal duties. The New Directions program will supply 50% of academic year salary up to $50,000 maximum. Applications must include a letter from the applicant's department chair indicating that the home institution will provide a minimum of 50% of academic year salary and all health and other relevant fringe benefits. (Deadline March 1, 2005.)

All IMA members are provided with an excellent and extremely stimulating research environment and connection with a large community of first class researchers. The IMA is a national institute whose mission is to increase the impact of mathematics by fostering research of a truly interdisciplinary nature, linking mathematics of the highest caliber and important scientific and technological problems from other disciplines and industry. Allied with this mission, the IMA also aims to expand and strengthen the talent base engaged in mathematical research applied to or relevant to such problems. It was founded in 1982 and receives its primary funding from the National Science Foundation.

Application forms and instructions are available at `http://www.ima.umn.edu/docs/membership.html`. More information on the IMA is available at `http://www.ima.umn.edu`, and informa-

tion on the 2005-2006 thematic program is at `http://www.ima.umn.edu/imaging`. Questions may be directed to `applications@ima.umn.edu` for postdoctoral fellowships and general membership applications or to `ndprof@ima.umn.edu` for New Directions professorships.

The University of Minnesota is an equal opportunity educator and employer.

Arnd Scheel, Deputy Director, IMA

Chairman's Column

In this issue's column I would like to take a few moments to describe some administrative matters that concern SIAG/OPT and its members. The first is the upcoming renewal of the interest group's charter. All SIAM interest groups have a charter that is renewed every three years, and SIAG/OPT will apply to have its charter renewed by the SIAM Board at the SIAM National Meeting in July, 2006. SIAM is in the process of implementing "uniform rules of procedure" for its various interest groups, and as part of this effort the charter for SIAG/OPT has been revised to bring it into line with the uniform format. The revised charter is available on SIAM's web pages at the URL `http://www.siam.org/activity/optimization/procedure.php`. There is one substantive change associated with this renewal. In addition to instituting the uniform rules of procedure, SIAM is making an effort to coordinate the terms of interest group officers with meetings that are sponsored by the interest groups, such as the triennial SIAM Optimization Conference sponsored by SIAG/OPT. In the case of SIAG/OPT, our meeting is currently "out of synch" with the terms of the officers. For example the current officers were elected to serve terms from 2004-2006, but the next Optimization Conference is in May, 2008. This is problematic since several officers have a role in the Optimization Conference; the SIAG Chair and Program Director are co-chairs for the conference, and the SIAG Vice-Chair is the chair of the committee that awards the SIAG/OPT Prize. This issue was discussed at the SIAG/OPT business meeting held during the Optimization Conference in Stockholm last year. At the business meeting a proposal was made to extend the terms of the current officers by one year to better coordinate the officers' terms with the triennial Optimization Conference. This proposal had broad support at the business meeting, was subsequently approved by SIAM, and is incorporated in the revised SIAG/OPT charter via the statement in Article IV that "The SIAG shall hold an election to fill those offices every 3 years, beginning in 2007".

The second issue I would like to mention is the location of the next Optimization Conference. As discussed at the business meeting in Stockholm, given the locations of the last two Optimization Conferences (Toronto and Stockholm) as well as the locations of the 2003 and 2006 Mathematical Programming Symposia (Copenhagen and Rio de Janeiro), SIAM had a preference for the 2008 Optimization Conference to be held at a site in the United States. I am pleased to announce that the 2008 SIAM Optimization Conference will be held from May 10-13 at the Park Plaza Hotel in Boston, Massachusetts. This location offers easy flight connections and excellent conference facilities. The hotel is attractively located near the Boston Common, and SIAM was able to negotiate an extremely reasonable hotel rate ($149 plus tax) for conference participants. The Boston Park Plaza is also the site for the 2006 SIAM Annual Meeting, but given the 2-year interval between the meetings, as well as the modest overlap in attendance, the conference organizers found the features of this site compelling. More details regarding the 2008 Optimization Conference will be forthcoming as the date of the meeting draws nearer.

Kurt M. Anstreicher, SIAG/OPT Chair
Department of Management Sciences
University of Iowa
S210 PBB Iowa City, IA 52242,
USA
kurt-anstreicher@uiowa.edu
`http://www.biz.uiowa.edu/faculty/anstreicher`