N^o 59

OPTIMA

Mathematical Programming Society Newsletter

OCTOBER 1998

Although pattern search methods were introduced forty years ago, they have recently been the subject of much renewed interest within the nonlinear programming community. For those of us who are new to the recent developments in the convergence theory for these methods, Virginia Torczon, Michael Lewis and Michael Trosset have prepared an overview on why these methods work.

–MARY BETH HRIBAR

Why Pattern Search Works*

Robert Michael Lewis
NASA Langley Research Center;
Virginia Torczon,
Michael W. Trosset
College of William & Mary

1 Introduction

Pattern search methods are a class of direct search methods for nonlinear optimization. Since the introduction of the original pattern search methods in the late 1950s and early 1960s [2, 5], they have remained popular with users due to their simplicity and the fact that they work well in practice on a variety of problems. More recently, the fact that they are provably convergent has generated renewed interest in the nonlinear programming community.

The purpose of this article is to describe what pattern search methods are and why they work. Much of our past work on pattern search methods was guided by a desire to unify a variety of existing algorithms and provide them with a common convergence theory. Unfortunately, the unification of this broad class of algorithms requires a technical framework that obscures the features that distinguish pattern search algorithms and make them work. We hope here to give a clearer explanation of these ideas. Space does not allow us to do justice to the history of these methods and all the work relating to them; this will be the subject of a lengthier review elsewhere; for a historical perspective, see [17].

*This research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-97046 while the authors were in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681-2199.

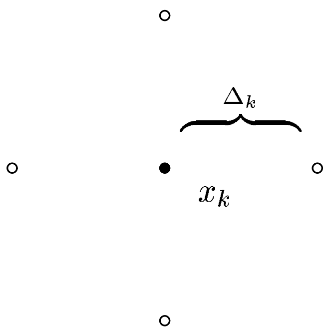


Figure 1. A simple instance of pattern search

2 A Simple Example of Pattern Search

We begin our discussion with a simple instance of a pattern search algorithm for unconstrained minimization: minimize $f(x)$. At iteration k, we have an iterate $x_k \in IR^n$ and a step-length parameter $\Delta_k > 0$. Let e_i, $i = 1, \ldots, n$, denote the standard unit basis vectors. We successively look at the points $x_+ = x_k \pm \Delta_k e_i$, $i = 1, \ldots, n$, until we find x_+ for which $f(x_+) < f(x_k)$. Fig. 1 illustrates the set of points among which we search for x_+ for $n = 2$. This set of points is an instance of what we call a pattern, from which pattern search takes its name. If we find no x_+ such that $f(x_+) < f(x_k)$, then we reduce Δ_k by a half and continue; otherwise, we leave the step-length parameter alone, setting $\Delta_{k+1} = \Delta_k$ and $x_{k+1} = x_+$. In the latter case we can also increase the step-length parameter, say, by a factor of 2, if we feel a longer step might be justified. We repeat the iteration just described until Δ_k is deemed sufficiently small.

This simple example illustrates two attractive features of pattern search algorithms:

- They can be extremely simple to specify and implement.
- No explicit estimate of the derivative nor anything like a Taylor's series appears in the algorithm. This makes these algorithms useful in situations where derivatives are not available and finite-difference derivatives are unreliable, such as when $f(x)$ is noisy.

These qualities have made pattern search algorithms popular with users. Yet, despite their seeming simplicity and heuristic nature and the fact that they do not have explicit recourse to the derivatives of $f(x)$, pattern search algorithms possess global convergence properties that are almost as strong as those of comparable line-search and trust-region algorithms. In this article we will attempt to explain this perhaps surprising fact.

Before turning to the discussion of how this can be, we note some further features of pattern search which are manifest in this simple example.

- We require only simple decrease in $f(x)$. In fact, we do not even need to know $f(x)$ as a numerical value, provided we can make the assessment that $f(x_+)$ is an improvement on $f(x_k)$.
- If we are lucky, we need only a single evaluation of $f(x)$ in any given iteration. Once we find an x_+ for which $f(x_+) < f(x_+)$, we can accept it and proceed. On the other hand, in the worst case we will look in quite a few directions ($2n$, for this example) before we try shorter steps.
- The steps that are allowed are restricted in direction and length. In this example, the steps must lie parallel to the coordinate axes and the length of any step has the form $\Delta_0/2^N$ for some integer N.

This simple example also suggests that there is a great deal of flexibility in pattern search algorithms, depending on how one specifies the pattern of points to be searched for the next iterate. These features will be recurring themes in our discussion.

3 The General Pattern Search Algorithm

For simplicity, our discussion will focus primarily on the case of unconstrained minimization,

$$\text{minimize} \quad f(x).$$

We assume that f is continuously differentiable, but that information about the gradient of f is either unavailable or unreliable. Since the inception of pattern search methods, various techniques have also been used to apply them to solve the general nonlinear programming problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & c(x) \le 0 \\ & \ell \le x \le u. \end{array}$$

More recently, pattern search methods specifically designed for constrained problems with an attendant convergence theory have been developed in [6, 9, 8].

The form of a general pattern search algorithm is quite simple and not all that different from any other nonlinear minimization algorithm: first, find a step s_k from the current iterate x_k; second, determine if that step is acceptable; and finally, update the critical components of the algorithm. At iteration k pattern search methods will consider steps in directions denoted by d_k. We require d_k to be a column of D_k, where D_k is an $n \times p_k$ matrix (i.e., D_k represents the set of directions under consideration).

Generalized pattern search:
Given $x_0 \in IR^n$, $f(x_0)$, $D_0 \in IR^{n \times p_0}$, and $\Delta_0 > 0$,
for $k = 0, 1, \ldots$ until done do {

1. Find a step $s_k = \Delta_k d_k$ using the procedure Exploratory_Moves Δ_k, (D_k).
2. If $f(x_k + \Delta_k d_k) < f(x_k)$, then $x_{k+1} = x_k + \Delta_k d_k$; otherwise, $x_{k+1} = x_k$.
3. Update (Δ_k, D_k)

}

In order to establish convergence results for this class of algorithms, we will, by and by, place additional conditions on D_k, the step calculation procedure Exploratory_Moves(), and the update procedure Update(). The analysis reveals that we do not need to explicitly define Exploratory_Moves() or Update(); for the purposes of ensuring convergence it suffices to specify conditions on the results they produce. We refer the interested reader to [16] for specific examples of Exploratory_Moves() and Update() used for some of the more traditional pattern search methods.

4 Global Convergence Analysis

Here we will use global convergence of an optimization algorithm to mean convergence to a stationary point of at least one subsequence of the sequence of iterates produced by the algorithm. A slightly weaker assertion is

$$\liminf_{k \to \infty} \| \nabla f(x_k) \| = 0;$$

this is equivalent to the previous property if the iterates $\{x_k\}$ remain in a bounded set.

Classical analyses of such methods as steepest descent and globalized Newton methods rely in a fundamental way on $\nabla f(x)$ to prove global convergence. Moreover, the technical conditions that make the proof of global convergence for these algorithms possible, such as the Armijo-Goldstein-Wolfe conditions for line-search methods, are actually built into the specification of gradient-based algorithms.

On the other hand, no such technical conditions appear in the description of pattern search algorithms (witness the example in §2). The philosophy of pattern search algorithms (and direct search methods in general) is best described by Hooke and Jeeves [5]:

> We use the phrase “direct search” to describe sequential examination of trial solutions involving comparison of each trial solution with the “best” obtained up to that time together with a strategy for determining (as a function of earlier results) what the next trial solution will be. The phrase implies our preference, based on experience, for straightforward search strategies which employ no techniques of classical analysis except where there is a demonstrable advantage in doing so[1].

This passage captures the basic philosophy of the original work on direct search algorithms: an avoidance of the explicit use or approximation of derivatives. Instead, the developers of the original direct search algorithms relied on heuristics to obtain what they considered promising search directions.

Nonetheless, we can prove global convergence results for pattern search methods, even though this class of algorithms was not originally developed with convergence analysis in mind. The analysis does ultimately rely on $\nabla f(x)$; hence the assumption that f is continuously differentiable. But because pattern search methods do not compute or approximate $\nabla f(x)$, the relationship between these algorithms and their convergence analysis is less direct than that for gradient-based algorithms.

4.1 The Ingredients of Global Convergence Analysis

We will now review the ideas that underlie the global convergence analysis of line-search methods for unconstrained minimization in order to compare them with those for pattern search. We focus on line-search methods rather than trust-region methods since the comparisons and contrasts with pattern search are simpler for line-search methods.

[1]It might strike the modern reader as odd that Hooke and Jeeves would question the advantages of employing techniques of “classical analysis” — meaning calculus — given the success of quasi-Newton algorithms. However, direct search methods appeared in the late 1950s and early 1960s, a time at which derivative-based methods were not as efficient as today, and no general convergence analysis existed for any practical optimization algorithm, derivative-based or not. The Armijo-Goldstein-Wolfe conditions [1, 4, 19], which form the basis for designing and analyzing what we now consider to be practical line-search algorithms, were several years in the future; trust region algorithms [14] were further still.

In order to prove global convergence of a line-search algorithm, at the very least one must show that if the current iterate x_k is not a stationary point, then the algorithm will eventually find an iterate x_{k+1} such that $f(x_{k+1}) < f(x_k)$. This unavoidably leads to the contemplation of the gradient, since the gradient ensures that a direction of descent can be identified: if x_k is not a stationary point of f, then any direction within 90˚ of $-\nabla f(x_k)$ is a descent direction. For our purposes, this will prove a crucial, if elementary, observation: one does not need to know the negative gradient in order to improve $f(x)$, one only needs a direction of descent. Then, if one takes a short enough step in that direction, one is guaranteed to find a point x_{k+1} such that $f(x_k+1) < f(x_k)$.

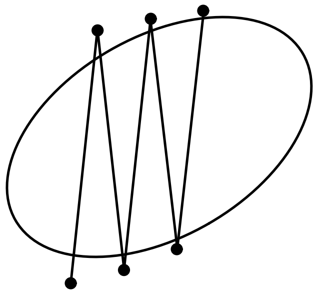


Figure 2: Decrease is too small relative to the length of the step

However, descent is not sufficient to ensure convergence: one must also rule out the possibility that the algorithm can simply grind to a halt, converging to a point that is not a stationary point. One begins by requiring at least one search direction to be uniformly bounded away from orthogonality with $-\nabla f(x_k)$. This ensures that the sequence of iterates cannot degenerate into steps along directions that become ever more orthogonal to the gradient while producing an ever diminishing improvement in $f(x)$.

This restriction on the search directions is still not sufficient to prevent the iterates from converging to points that are not stationary points. This unhappy situation can occur in two ways. First, there is the pathology depicted in Fig. 2. The ellipse represents a level set of $f(x)$, which in this case is a convex quadratic. The steps taken are too long relative to the amount of decrease between successive iterates.

While the sequence of iterates $\{x_k\}$ produces a strictly decreasing sequence of objective values $\{f(x_k)\}$, the sequence of iterates converges to two nonstationary points.

The other pathology, depicted in Fig. 3, occurs when the amount of decrease between successive iterates is too small relative to the amount of decrease initially seen in the direction from one iterate to the next. This time the steps between successive iterates become excessively short. This sequence converges to a single point which again is not a stationary point.

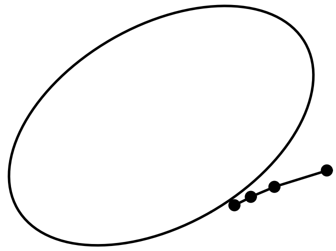


Figure 3: Decrease is too small relative to the norm of the gradient

These pathologies lead to the second standard element of global convergence analysis: a mechanism that controls the length of the step. Both of the preceding pathologies can be avoided, for instance, by requiring that the amount of decrease in $f(x)$ between successive iterates be “sufficient,” where sufficient relates the amount of decrease, the length of the step, and the gradient $\nabla f(x)$. This is the purpose of the Armijo-Goldstein-Wolfe conditions for line-search algorithms: given a suitable descent direction d_k, we choose a step length $\Delta_k > 0$ such that for some fixed $\alpha \in (0,1)$ and fixed $\beta \in (\alpha,1)$, $x_{k+1} = x_k + \Delta_k d_k$ satisfies both

$$f(x_{k+1}) \le f(x_k) + \alpha \Delta_k \nabla f(x_k)^T d_k \qquad (1)$$

and

$$\nabla f(x_{k+1})^T d_k \ge \beta \nabla f(x_k)^T d_k. \qquad (2)$$

The first condition precludes steps that are too long; the second condition precludes steps that are too short.

5 How Pattern Search Does Its Thing

We can summarize the devices that ensure the global convergence of line-search methods for unconstrained minimization as follows:

1. The choice of a suitably good descent direction.
2. Step-length control:
   - (a) a mechanism to avoid steps that are too long, and
   - (b) a mechanism to avoid steps that are too short, where long and short refer to the sufficient decrease conditions (1) and (2), respectively.

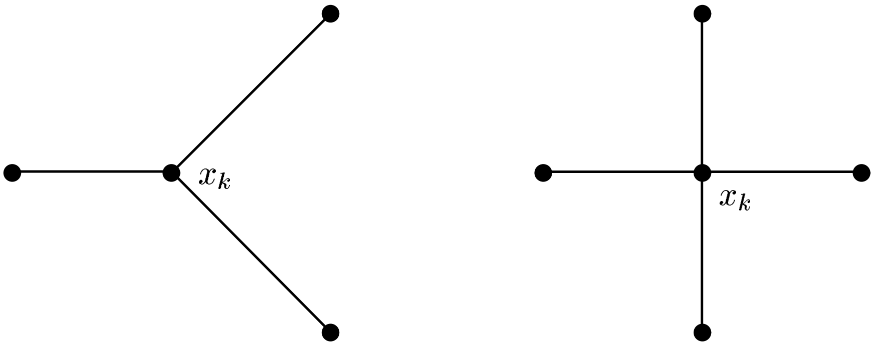


Figure 4: Examples of a minimal and a maximal positive basis for R^2

These mechanisms, which are explicitly built into line-search algorithms, all depend on information about the gradient. However, pattern search algorithms do not assume such information, and thus do not and cannot enforce such conditions. What, then, ensures the global convergence of pattern search algorithms?

The answer resembles the classical arguments for establishing the global convergence of line-search methods, but necessarily with novel elements. As we shall see, pattern search algorithms are globally convergent because:

1. At each iteration, they look in enough directions to ensure that a suitably good descent direction will ultimately be considered.
2. They possess a reasonable back-tracking strategy that avoids unnecessarily short steps.
3. They otherwise avoid unsuitable steps by restricting the nature of the step allowed between successive iterates, rather than by placing requirements on the amount of decrease realized between successive iterates.

At the heart of the argument lies an unusual twist: we relax the requirement of sufficient decrease and require only simple decrease ($f(x_{k+1}) < f(x_k)$), but we impose stronger conditions on the form the step s_k may take. Furthermore, this trade-off is more than just a theoretical innovation: in practice, it permits useful search strategies that are precluded by the condition of sufficient decrease.

5.1 Pattern Search as a Crypto-Gradient Method

The analysis begins by demonstrating that a search direction not too far from the negative gradient is always available. This is accomplished by considering a set of step directions D_k sufficiently rich that it necessarily includes at least one acceptable descent direction. In the absence of any estimate of $-\nabla f(x_k)$, pattern search algorithms hedge against the fact that $-\nabla f(x_k)$ could point in any direction.

For the example in §2 the set of directions D_k is $\{\pm e_i, i = 1, \ldots, n\}$, so the set of prospective next iterates has the simple form $\{x_k \pm \Delta_k e_i, i = 1, \ldots, n\}$. If a step $s_k = \pm \Delta k\, e_i$ producing simple decrease on $f(x_k)$ is found, then $x_{k+1} = x_k \pm \Delta k\, e_i$; otherwise, reduce Δ_k and try again. Other of the original pattern search methods, such as the method of Hooke and Jeeves [5] or coordinate search [13], also include in D_k the directions $\{\pm e_i, i = 1, \ldots, n\}$.

The analysis in [16] allows for more general conditions on the set of directions. In particular, D_k must contain a set of the form $\{\pm p_i, i = 1, \ldots, n\}$, where $p_1, \ldots, p_n$ is any linearly independent set of vectors. One can allow this set to vary with k, so long as one restricts attention to a finite collection of such sets.

The discussion in [18] brought to our attention that even less is required: it suffices that the set of directions D_k contain a positive basis Π_k for IR^n [7]. In terms of the theory of positive linear dependence [3], the positive span of a set of vectors $\{a_1,\ldots,a_r\}$ is the cone

$$\{a \in IR^n \mid a = c_1a_1 + \ldots + c_ra_r,\ c_i \geq 0 \text{ for all } i\}.$$

The set $\{a_1,\ldots,a_r\}$ is called positively dependent if one of the a_i's is a nonnegative combination of the others; otherwise the set is positively independent. A positive basis is a positively independent set whose positive span is IR^n, i.e., a set of generators for a cone that happens to be a vector space. A positive basis must contain at least n+1 vectors and can contain no more than 2n vectors [3]; we refer to the former as minimal and the latter as maximal; Fig. 4 demonstrates examples of both for R^2.

How do we know that at least one of the directions in D_k is not too orthogonal to the direction of steepest descent, regardless of what $-\nabla f(x)$ might be? A proof by picture is given in Fig. 5; see [7] for details. Consider the minimal positive basis $\{(1,1)^T, (1,-1)^T, (-1,0)^T\}$ depicted in Fig. 5 as directions emanating from x_k. Notice that the angles between these vectors are 90°, 135°, and 135°. For *any* continuously differentiable function $f: IR^2 \rightarrow IR$, if x_k is not a stationary point, then $-\nabla f(x_k)$ can be no more than 67.5° from one of the vectors in the positive basis, as shown in Fig. 5. Thus, including a positive basis Π_k in the set of directions D_k guarantees that we can approximate the negative gradient in a way that cannot be arbitrarily bad. This is the first step towards establishing global convergence.

5.2 The Underlying Lattice Structure

As it happens – as it was meant to happen – pattern search methods are restricted in the nature of the steps they take. This ultimately turns out to be the reason pattern search methods can avoid the pathologies illustrated in Fig. 2 and Fig. 3 without enforcing a sufficient decrease condition.

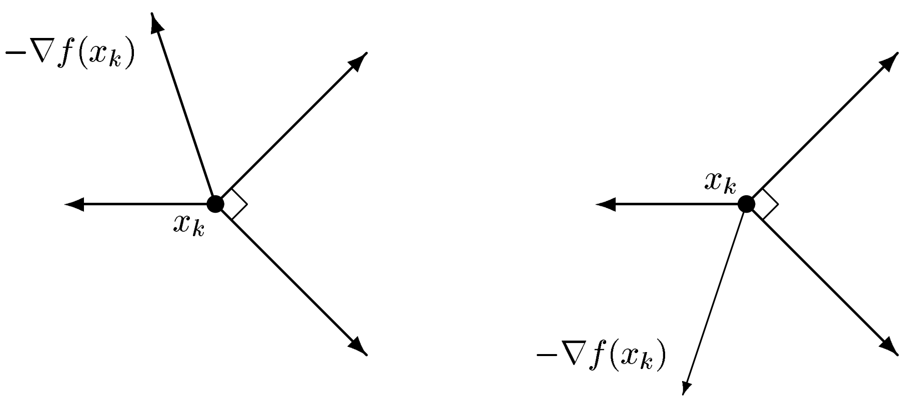


Figure 5: A minimal positive basis for R^2 and the two worst cases for $-\nabla f(x_k)$

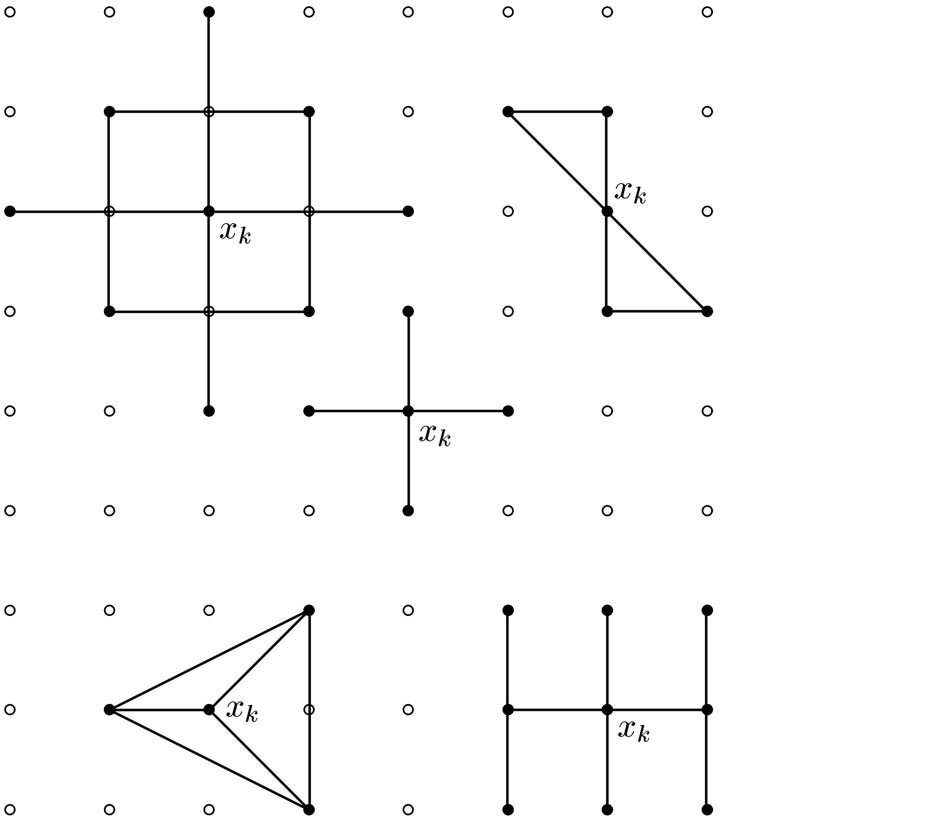


Figure 6: Some possible patterns

Let P_k denote the set of candidates for the next iterate (i.e., $P_k = x_k + \Delta_k D_k$, by abuse of notation). We call P_k the *pattern*, from which pattern search takes its name. Several traditional patterns are depicted in Fig. 6.

Though it does not appear to have been by any conscious design on the part of the original developers of pattern search algorithms, these algorithms produce iterates that lie on a finitely generated rational lattice, as indicated in Fig 6. More precisely, there exists a set of generators g_1, …, g_m, independent of k, such that any iterate x_k can be written as

$$x_k = x_0 + \Delta_0 \sum_{i=1}^{m} c_i^k g_i \qquad (3)$$

where each c_i^k is rational.

Note that this structural feature means that the set of possible steps, and thus the set of possible iterates, is known in advance and is independent of the actual objective $f(x)$. This is in obvious contrast to gradient-based methods.

Furthermore – and this is significant to the convergence analysis of pattern search – by judicious (but not especially restrictive) choice of the factors by which Δ_k can be increased or decreased, we can establish the following behavior of these algorithms. Suppose the set $\{x \mid f(x) \leq f(x_0)\}$ is bounded. Given any $\Delta_* > 0$, there exists a finite subset (that depends on Δ_*) of the lattice of all possible iterates such that x_k must belong to this subset until $\Delta_k < \Delta_*$. That is, there is only a finite number of distinct values x_k can possibly have until such time as $\Delta_k < \Delta_*$. This means that the only way to obtain an infinite sequence of distinct x_k is to reduce the step length parameter Δ_k infinitely often so that $\liminf_{k \to \infty} \Delta_k = 0$.

This also reveals another role played by the parameter Δ_k. Reducing Δ_k increases the set of candidate iterates by allowing us to search over a finer subset of the rational lattice of all possible iterates. This is shown in Fig. 6 and Fig. 7. In these pictures, halving Δ_k refines the grid over which we are tacitly searching for a minimizer of $f(x)$, while halving the minimum length a step is allowed to have.

5.3 Putting It All Together

We return to the general pattern search algorithm:

Generalized pattern search:

Given $x_0 \in IR^n$, $f(x_0)$, $D_0 \in IR^n \times p_0$, and $\Delta_0 > 0$, for $k = 0, 1, \ldots$ until done do {

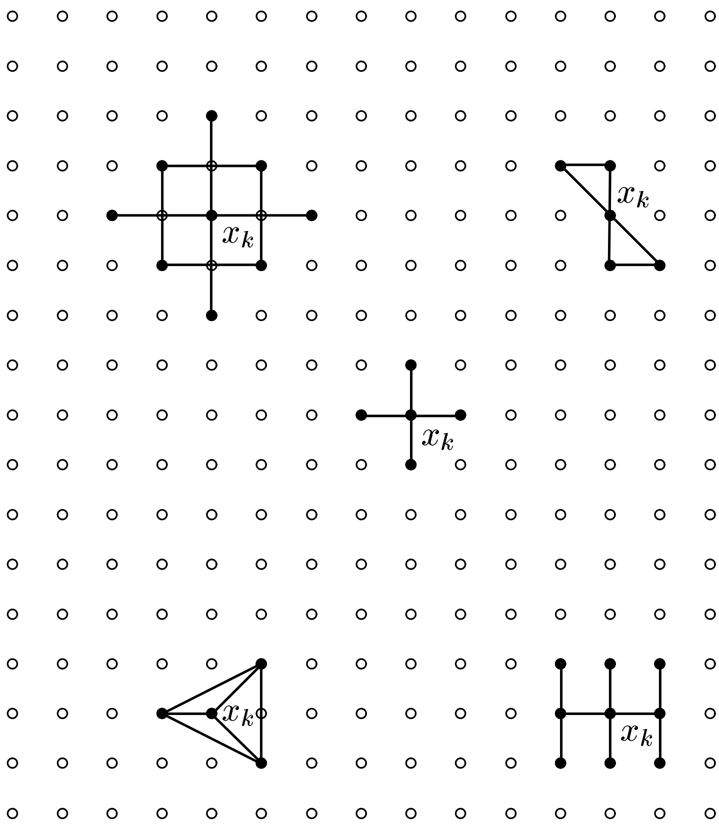


Figure 7: The same patterns on a refinement of the grid

1. Find a step $s_k = D_k\, d_k$ using the procedure Exploratory_Moves (D_k, D_k).
2. If $f(x_k + D_k\, d_k) < f(x_k)$, then $x_{k+1} = x_k + D_k\, d_k$; otherwise, $x_{k+1} = x_k$.
3. Update (D_k, D_k)

}

The step s_k returned by the Exploratory_Moves() algorithm must satisfy two simple conditions:

1. The step returned must be an element of $D_k D_k$.
2. The step s_k must satisfy either $f(x_k + s_k) < f(x_k)$ or $s_k = 0$.

Furthermore, s_k may be 0 only if none of the steps in $D_k\, P_k$ yielded decrease on $f(x_k)$.
The first condition prevents arbitrary steps along arbitrary directions; the second condition is a back-tracking control mechanism that prevents us from taking shorter steps unless it is truly necessary.

As for the procedure Update() for D_k and D_k, we are free to make modifications to D_k before the next iteration. Classical pattern search methods typically specify a single $D = D_k$ for all k. Others make substantive changes in response to the outcome of the exploratory moves. This is just one of many options to consider when designing a pattern search method, and it leads to a great deal of flexibility in this class of algorithms. There are conditions that must be satisfied to preserve the lattice structure, but these are straightforward to satisfy in practice. The interested reader is referred to [7], for a complete discussion of the technical conditions, and to [16], for a description of some traditional choices.

The rules for updating D_k are also restricted by the need to preserve the algebraic structure of the possible iterates. Historically, the popular choices have been to halve D_k at unsuccessful iterations, and to either leave D_k alone at successful iterations or possibly double it. The convergence analysis leads to other possibilities: we can rescale D_k by $Q \in \{t^{w_0}, \ldots, t^{w_L}\}$, where t is a rational number, $\{w_0, \ldots, w_L\}$ are integers, $L \geq 2$, $w_0 < 0$ and $w_L \geq 0$. This provides at least one option for reducing D_k when back-tracking is called for, and at least one option that does not reduce D_k.

The proof of convergence now goes like this. Suppose x_k is not a stationary point of $f(x)$. Because at least one of the directions d_k in P_k is necessarily a descent direction, we can always find an acceptable step $D_k d_k$ once we reduce D_k sufficiently. Thus, we can always find x_{k+1} with $f(x_{k+1}) < f(x_k)$ for k in some subsequence K.

Now, if $\liminf_{k \to \infty} \| \nabla f(x_k) \| \neq 0$, then for some $e > 0$, $\| \nabla f(x_k) \| > e$ for all k. Under this assumption we can show that once D_k is sufficiently small relative to e, it will no longer be reduced. This is so because one of the directions d_k in P_k is sufficiently close to $-\nabla f(x_k)$ to be a uniformly good descent direction, and $\| \nabla f(x_k) \|$ is uniformly not too small, so we will have $f(x_k + D_k d_k) < f(x_k)$ without having to drive D_k to zero.

However, if $\liminf_{k \to \infty} D_k = D_* > 0$, then due to the lattice structure of the iterates, there can be only finitely many possible x_k, contradicting the fact that we have an infinite subsequence K with $f(x_{k+1}) < f(x_k)$ for all $k \in K$ (assuming $\{x | f(x) \leq f(x_0)\}$ is bounded). Hence $\liminf_{k \to \infty} \| \nabla f(x_k) \| = 0$.

The correlation between the fineness of the grid of possible iterates and the size of D_k also explains why long steps are not a problem. We have argued above that if $\liminf_{k \to \infty} D_k = 0$, then $\liminf_{k \to \infty} \| \nabla f(x_k) \| = 0$. Now, unless $\liminf_{k \to \infty} D_k = 0$, there can be only a finite number of distinct iterates, and hence only a finite number of long steps (or any type of step, for that matter). Thus even if an infinite number of "bad" long steps are taken (i.e., steps that decrease $f(x)$ but that violate (1)), the mere fact that there are infinitely many distinct iterates means that $\liminf_{k \to \infty} D_k = 0$, and hence $\liminf_{k \to \infty} \| \nabla f(x_k) \| = 0$.

5.4 Observations

This analysis might suggest an interpretation of pattern search as a search over successively finer finite grids. If the finite set of candidates is exhausted without finding a point that improves $f(x)$, then the grid is refined by reducing D_k and the process is repeated.

However, this interpretation is misleading insofar as it suggests that pattern search algorithms are exceedingly inefficient. In practice, pattern search algorithms do not resort to searching over all the points in increasingly fine grids but instead behave more like a steepest descent method. In this sense, the analysis does not reflect the actual behavior of the algorithm. This should not be entirely surprising since, unlike gradient-based methods, the specification of pattern search algorithms does not obviously contain a mechanism designed to guarantee convergence.

The situation is analogous to that of the simplex method in linear programming. Once one establishes that the simplex method cannot cycle, the convergence of the algorithm follows from the fact that there is only a finite number of vertices that the simplex method can visit in its search for a solution. This means that the simplex method could and does have a theoretical worst-case complexity that is exponential, but in practice the simplex method has proven much more efficient than that.

Moreover, the actual behavior of pattern search in any single iteration can be very different than the proof of convergence might be thought to suggest. The search can accept as the next iterate *any* point in P_k that satisfies the simple decrease condition $f(x_{k+1}) < f(x_k)$. In particular, the algorithm does not necessarily need to examine every point in $D_k P_k$; it need only do so before deciding to reduce D_k, which is the worst case.

In the best case, we may need only a single evaluation of $f(x)$ to find an acceptable step. In contrast, in a forward-difference gradient-based method one needs at least n+1 evaluations of $f(x)$ (in addition to $f(x_k)$) to find a new iterate; n additional values of $f(x)$ to approximate $\nabla f(x_k)$ and at least one more evaluation of $f(x)$ to decide whether or not to accept a new iterate.

In order to make progress, pattern search requires the eventual reduction of D_k. The cost of discovering the necessity of this step is one evaluation of $f(x)$ for each direction defined by the positive basis P_k. For a minimal positive basis of n+1 elements, this cost is the same as the cost of an unsuccessful quasi-Newton step using a forward-difference approximation of the gradient; n evaluations of $f(x)$ to form the finite-difference approximation to $\nabla f(x_k)$, and the evaluation of $f(x)$ at the rejected x_+. On the other hand, following an unsuccessful step in the latter algorithm, one gets to reuse the gradient approximation; it is not clear how best to reuse information from unsuccessful iterations of pattern search in subsequent iterations.

5.5 The Resulting Convergence Results

Let $\mathcal{L} = \{x \mid f(x) \leq f(x_0)\}$, and suppose f is C^1 on a neighborhood of $\mathcal{L}$.

Theorem If $\mathcal{L}$ is bounded, then the iterates produced by a pattern search algorithm satisfy

$$\liminf_{k \to \infty} \| \nabla f(x_k) \| = 0.$$

If, in addition, $\lim_{k \to \infty} D_k = 0$ and we require $f(x_{k+1}) < f(x_k + s_k)$ for all $s_k \in D_k P_k$, the steps associated with the positive basis, and the columns of D_k are bounded in norm uniformly in k, then we have

$$\lim_{k \to \infty} \| \nabla f(x_k) \| = 0.$$

By way of comparison, we obtain the result $\lim_{k\ \mathrm{fi}}\ \|\ f(x_k)\| = 0$ for line-search methods without the assumption that is bounded [12]. However, we must also require sufficient decrease between iterates according to (1)–(2), rather than just simple decrease, that is, $f(x_{k+1}) < f(x_k)$.

For trust-region methods, with the assumption that $f(x)$ is uniformly continuous (but again, without the assumption that is bounded), requiring only simple decrease $f(x_{k+1}) < f(x_k)$ suffices to prove that $\lim_{k\ \mathrm{fi}}\ \|\ f(x_k)\| = 0$, provided the approximation of the Hessian does not grow too rapidly in norm [15]. With a sufficient decrease condition, one obtains the stronger result [11], $\lim_{k\mathrm{fi}}\ \|\ f(x_k)\| = 0$. However, for either result $f(x)$ is used in both the fraction of Cauchy decrease condition on the step and the update of the trust radius.

Thus, under the hypothesis that is bounded, the global convergence results for pattern search algorithms are as strong as those for gradient-based methods. This might seem surprising, but it simply reflects just how little one needs to establish global convergence. Pattern search is sufficiently like steepest descent that it works.

This leads to one caveat for users: like steepest descent, pattern search methods are good at improving an initial guess and finding a neighborhood of a local solution, but fast local convergence should not be expected. In general, one can expect only a linear rate of local convergence.

6 Concluding Remarks

We have tried to explain how and why pattern search works while refraining from a detailed description of the convergence analysis. Once one understands the essential ideas, the proof of global convergence is reasonably straightforward, if sometimes tedious. Precisely because pattern search methods have so little analytical information explicitly built into them, it takes some effort to extract an assurance that they actually do work. However, as we have tried to indicate, many of the ideas are familiar from standard analysis of nonlinear programming algorithms. The novelty lies in the restriction of the iterates to a lattice, which allows us to relax the conditions on accepting steps.

The ideas discussed here also appear in the analysis of pattern search methods for constrained minimization [6, 9, 8]. For readers who would like to explore the connections between pattern search methods and gradient-based algorithms in greater detail, we particularly recommend [10].

References

[1] L. Armijo, *Minimization of functions having Lipschitz continuous first partial derivatives*, Pacific Journal of Mathematics, 16 (1966), pp. 1-3.

[2] G.E.P. Box, *Evolutionary operation: A method for increasing industrial productivity*, Applied Statistics, 6 (1957), pp. 81-101.

[3] C. Davis, *Theory of positive linear dependence*, American Journal of Mathematics, 76 (1954), pp. 733-746.

[4] A.A. Goldstein, *Constructive Real Analysis*, Harper & Row, New York, NY, 1967.

[5] R. Hooke and T.A. Jeeves, *Direct search solution of numerical and statistical problems*, Journal of the Association for Computing Machinery, 8 (1961), pp. 212-229.

[6] R.M. Lewis and V.J. Torczon, *Pattern search algorithms for bound constrained minimization*, Tech. Rep. 96-20, Institute for Computer Applications in Science and Engineering, Mail Stop 403, NASA Langley Research Center, Hampton, Virginia 23681-2199, March 1996. To appear in SIAM Journal on Optimization.

[7] ——, *Rank ordering and positive bases in pattern search algorithms*, Tech. Rep. 96-71, Institute for Computer Applications in Science and Engineering, Mail Stop 403, NASA Langley Research Center, Hampton, Virginia 23681-2199, 1996. In revision for Mathematical Programming.

[8] ——, *A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds*, Tech. Rep. 98-31, Institute for Computer Applications in Science and Engineering, Mail Stop 403, NASA Langley Research Center, Hampton, VA 23681-2199, July 1998. Submitted to SIAM Journal on Optimization.

[9] ——, *Pattern search methods for linearly constrained minimization*, Tech. Rep. 98-3, Institute for Computer Applications in Science and Engineering, Mail Stop 403, NASA Langley Research Center, Hampton, Virginia 23681-2199, January 1998. To appear in SIAM Journal on Optimization.

[10] S. Lucidi and M. Sciandrone, *On the global convergence of derivative free methods for unconstrained optimization*, Tech. Rep. 18-96, DIS, Universitá di Roma "La Sapienza," 1996. Submitted to SIAM Journal on Optimization.

[11] J.J. Moré, *The Levenberg-Marquardt algorithm: implementation and theory*, in Numerical Analysis: Proceedings of the biennial conference, Dundee 1977, Lecture Notes in Mathematics no. 630, G. A. Watson, ed., Springer-Verlag, Berlin, 1978, pp. 105-116.

[12] S.G. Nash and A. Sofer, *Linear and Nonlinear Programming*, McGraw-Hill, New York, 1996.

[13] E. Polak, *Computational Methods in Optimization: A Unified Approach*, Academic Press, New York, 1971.

[14] M.J.D. Powell, *A new algorithm for unconstrained optimization*, in Nonlinear Programming, J. B. Rosen, O. L. Mangasarian, and K. Ritter, eds., Academic Press, New York, NY, 1970, pp. 31-65.

[15] ——, *Convergence properties of a class of minimization algorithms*, in Nonlinear Programming 2, O.L. Mangasarian, R.R. Meyer, and S.M. Robinson, eds., Academic Press, New York, NY, 1975, pp. 1-27.

[16] V. Torczon, *On the convergence of pattern search algorithms*, SIAM Journal on Optimization, 7 (1997), pp. 1-25.

[17] V. Torczon and M.W. Trosset, *From evolutionary operation to parallel direct search: Pattern search algorithms for numerical optimization*, Computing Science and Statistics, 29 (1998), pp. 396-401.

[18] Y. Wen-ci, *Positive basis and a class of direct search techniques*, Scientia Sinica, Special Issue of Mathematics, 1 (1979), pp. 53-67.

[19] P. Wolfe, *Convergence conditions for ascent methods*, SIAM Review, 11 (1969), pp. 226-235.

Robert Michael Lewis, Institute for Computer Applications in Science and Engineering (ICASE), Mail Stop 403, NASA Langley Research Center, Hampton, Virginia 23681-2199, U.S.A., buckaroo@icase.edu

Virginia Torczon, Department of Computer Science, College of William & Mary, P.O. Box 8795, Williamsburg, Virginia 23187-8795, U.S.A., va@cs.wm.edu.

Michael W. Trosset, College of William & Mary, P.O. Box 8795, Williamsburg, Virginia 23187-8795, U.S.A., trosset@math.wm.edu

October 12, 1998

Farewell Remarks by the Outgoing MPS Chair

In August 1998, my term as Chair of the Mathematical Programming Society ended. I confess to some relief and some nostalgia. It is the greatest honor of my career to have won election as Chair of the major professional organization for my primary research interest, and I have enjoyed the opportunity to make a difference. It has been a busy three years. My officers and I have made some significant changes that will benefit the Society, and Tom Liebling and his Lausanne Organizing Committee put on a magnificent Symposium during our watch.

My successor, Jean-Philippe Vial, inherits a financially sound society in the process of digesting these changes, but he also inherits a society whose membership has declined by about 10% over the last three years. This problem was the result of an oversight in the registration process for the Lausanne meeting. Nonmembers were not offered membership at a reduced rate. Please support the upcoming membership drive by renewing your membership. If you are not a member, then you must be reading this on the MPS web site, and you can join at this site.

We have made two major changes. In a move that I believe is crucial to the continued health of the MPS, we have changed publishers for the journals, MP A&B. The key to this change was our desire to reduce the library subscription price of the journals while keeping member subscription as a membership benefit.

Many of you have told us of battles with cost conscious librarians to keep our journals in your libraries. The price will roughly halve next year. Springer will publish the series A&B under the same titles and with only a slightly modified cover art. Our feeling is that allowing Springer to make a slight change in the cover signals the changes without leading to any confusion that might lead a librarian to think that this is a new journal.

The second move is that SIAM is now handling our member services. We are receiving more services and more reliable services with no increase in cost. For a while, it may not have seemed that way because many renewals went awry. This happened because many of you seemed not to have gotten the cover letter with the ISI mailing of renewal invoices, asking you to renew through SIAM. Still, SIAM seems to have straightened out this problem, even though they had no hand in causing it. In addition, the SIAM staff made a real effort to make our directory accurate.

Of course, we all knew that the previous directories were unusable, but I did not realize, until the new directory came out and I saw the volume of mail generated to correct errors, that ISI seems not to have made changes to the directory even when members notified them. I am confident that the current directory is useful and that next year our directory will be very accurate. We can help this process by providing accurate e-mail addresses on our renewal forms.

Elsewhere in this issue of OPTIMA, Jean-Philippe Vial will write the column that signals the beginning of his term. He will do a fine job, and he will do it with his usual sense of style. Let us resolve to help him all we can.

Finally, let me thank you again for the opportunity to serve the MPS in this capacity, and let me also thank the Council and the Executive Committee chaired heroically by Steve Wright for being partners this term. Vice-Chairs Jan Karel Lenstra and Jean-Philippe Vial, as well as Treasurer Clyde Monma were all I could have wished and more.

–JOHN DENNIS

The New MPS Chair

When I started to learn optimization and OR, I was motivated by the belief that quantitative decision-making in business and organizations was a field full of promise. The belief was naive, or at least not based on proper information. The decades that followed did not fulfill these great expectations. Our field certainly has shown much vitality and creativity and some spectacular applications were achieved, but we must also acknowledge certain failures: unrealistic models, insufficient databases, lack of appropriate software, and drastic hardware limitations. As a result, the business community developed much skepticism about our profession. Optimization and OR requirements diminished or disappeared from MBA curricula at many institutions.

Those bad times seem to be over. I am happy to be starting my duties of Chair at a time of bright prospects for our profession. It is commonplace to credit the hardware revolution as a major cause, but in addition we should note that algorithms progressed at an even greater rate. A third factor may, however, be the one that brings us real support from the business community: Spreadsheets and modeling languages nowadays provide an environment in which the users can think and express their problem. They free users from most of the mathematical intricacies and subtleties that we cherish and thereby encourage them to explore new ways of thinking based on models and optimization.

It should be one of our goals to regain our popularity in the business community through a new approach to teaching and consulting. The potential for applications is enormous. In some areas, decision-makers have no other alternative than the best methodology we can provide. This is certainly true of combinatorial problems that are so easy to formulate and so difficult to solve. But I also have in mind some strategic issues, such as measuring the economic impact of an effort to reduce greenhouse gas emission, estimating traffic congestion, or evaluating oligopolistic situations in a deregulated world. In these examples, the concept of equilibrium is probably the only one that gives a grasp to the analysis, though computing equilibria efficiently still remains a challenge in many instances.

Equally important for the future of our Society is our ability to promote new applications in engineering. As our former chair pointed out three years ago, engineers often ignore the availability of powerful optimization tools that would improve the design and the control of engineering systems. Optimization is still not fully part of the engineering culture, even though the minimum energy principle is so basic and omnipresent in physics. The failure may be due to insufficient performance of nonlinear solvers in the past. It may also be that engineers are more interested in the design of reliable and robust systems than in obtaining the ultimate with respect to some criterion. We should publicize optimization not as the way to get “the” solution, but rather as an intelligent simulation tool. Optimization often leads to novel and sometimes surprising solutions that contribute to a better knowledge and mastering of engineering systems. The new field of robust optimization also offers great opportunities.

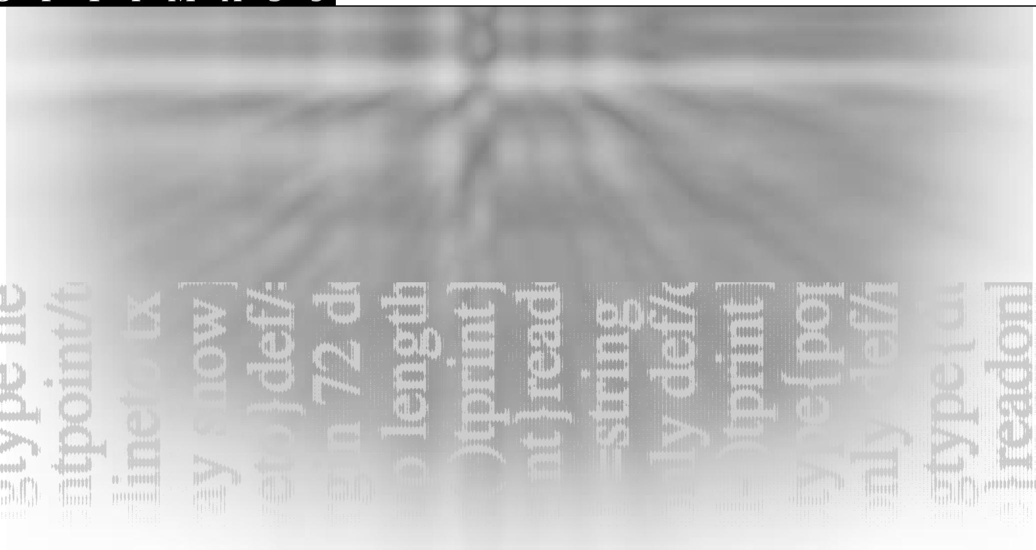
Having a positive attitude towards applications does not mean that we should neglect the research-oriented character of our Society. Development of new theories and new algorithms is essential to continued vitality of the field, and remains the primary goal of the Society’s communications media, in particular, our journals and symposia. Although most of us are not practitioners, making optimization a more and more operational tool should also become our concern. To maintain a proper balance between those two facets of our activity is our challenge. We may be helped in this mission by cooperation with sister societies SIAM, INFORMS, IFORS, which have related but different foci.

The previous team initiated some major management innovations, including a new publisher for Mathematical Programming with a lower institutional subscription rate, and a new membership services provider. The change of publishers to Springer will take effect next January, and everyone should feel responsible for making their own organization’s library aware of the new lower rate and encouraging them to subscribe. Electronic distribution is also part of the new publisher’s program. Membership services have been provided by the SIAM office since the start of 1998. They have been working hard to update the data base of members (which, unfortunately, had not been maintained well for some years) and to enhance our Internet presence. Electronic searching of the database is now possible through the MPS web site (http://www.mathprog.org/). In fact, most of your contacts with MPS, including correcting your membership information, renewing membership, pursuing OPTIMA online, and gathering information on MPS prizes and upcoming symposia, can be performed through the web site.

The election took place this year and brought in a new council. Steve Wright (wright@mcs.anl.gov) continues to hold the appointed post of Executive Committee Chair, while the Publications Committee continues to be chaired by Bob Bixby. Our immediate task is to achieve larger diffusion of our journal and to increase membership. Above all, we should maintain the unique character of the Society, most notably its high scientific level and its genuine international flavor. We are particularly looking forward to the meeting in Atlanta in the year 2000, and the Search Committee is actively looking for an attractive and exciting place in 2003.

Please contact me (at chair@mathprog.org or jean-philippe.vial@hec.unige.ch), Steve Wright, or the Executive Committee (at xcom@mathprog.org) with any comments on Society business. The low periodicity of Society gatherings favors their quality, but unfortunately, it gives us limited opportunities to exchange views on MPS. Electronic mail partially compensates for this lack of personal contact, so please don’t hesitate to use it and let us know what you think.

–Jean-Philippe Vial
University of Geneva, 102 Bd Carl Vogt,
CH 1211 Geneva 4, Switzerland;
Office: +41 22 705 81 24,
Fax: +41 22 705 81 04;
e-mail: jean-philippe.vial@hec.unige.ch



Conference Calendar

▶ **International Conference on Nonlinear Programming and Variational Inequalities**
December 15-18, 1998, Hong Kong
URL: http://www.cityu.edu.hk/ma/conference/icnpvi/icnpvi.html

▶ **DIMACS Conference on Semidefinite Programming and Large Scale Discrete Optimization**
January 7-9, 1999, Princeton University
URL: http://dimacs.rutgers.edu/Workshops/SemidefiniteProg/index.html

▶ **DIMACS Conference on Algorithm Engineering and Experimentation**
January 15-16, 1999, Baltimore, MD
URL: http://dimacs.rutgers.edu/Workshops/Algorithm/

▶ **Tenth Annual ACM-SIAM Symposium on Discrete Algorithms**
January 17-19, 1999, Baltimore, Maryland
URL: http://www.siam.org/meetings/da99/

▶ **DIMACS Conference on Large Scale Discrete Optimization in Manufacturing and Transportation**
February 8-10, 1999, DIMACS Center Rutgers

▶ **DIMACS Conference on Mobile Networks and Computing**
March 24-26, 1999, DIMACS Center Rutgers
URL: http://dimacs.rutgers.edu/Workshops/Mobile/index.html

▶ **INFORMS National Meeting**
May 2-5, 1999, Cincinnati, KY
URL: http://www.cba.uc.edu/dept/qa/cinforms/

▶ **Sixth SIAM Conference on Optimization**
May 10-12, 1999, Atlanta, GA
URL: http://www.siam.org/meetings/op99/index.htm

▶ **1999 SIAM Annual Meeting**
May 12-15 1999, Atlanta, GA
URL: http://www.siam.org/meetings/an99/index.htm

▶ **Workshop on Continuous Optimization**
June 21-26, 1999, Rio de Janeiro
URL: http://www.impa.br/~opt/

▶ **Fourth International Conference on Industrial and Applied Mathematics**
July 5-9, 1999, Edinburgh, Scotland
URL: http://www.ma.hw.ac.uk/iciam99/

▶ **19th IFIP TC7 Conference on System Modelling and Optimization**
July 12-16, 1999, Cambridge, England
URL: http://www.damtp.cam.ac.uk/user/na/tc7con

Call for Papers

Seventh Conference on Integer Programming and Combinatorial Optimization

IPCO '99
June 9-11, 1999
TU Graz, Graz, Austria

Conference Approach

This meeting, the seventh in the series of IPCO conferences held every year in which no MPS International Symposium takes place, is a forum for researchers and practitioners working on various aspects of integer programming and combinatorial optimization. The aim is to present recent developments in theory, computation, and applications of integer programming and combinatorial optimization. Topics include, but are not limited to: polyhedral combinatorics; integer programming; cutting planes; branch and bound; geometry of numbers; semidefinite relaxations; matroids and submodular functions; computational complexity; graph and network algorithms; approximation algorithms; on-line algorithms; and scheduling theory and algorithms.

In all these areas, we welcome structural and algorithmic results, revealing computational studies, and novel applications of these techniques to practical problems. The algorithms studied may be sequential or parallel, deterministic or randomized.

During the three days, approximately 36 papers will be presented in a series of sequential (non-parallel) sessions. Each lecture will be 30 minutes long. The Program Committee will select the papers to be presented on the basis of extended abstracts to be submitted as described below.

The proceedings of the conference will be published in the Springer Lecture Notes in Computer Science series and will contain full texts of all presented papers. Copies will be provided to all participants at registration time.

Paper Submission

An extended abstract (up to 10 pages) must be submitted by November 15, 1998. Electronic submissions (in PostScript) are strongly encouraged. Please refer to the Conference web site for further submission instructions.

If an electronic submission is not possible, submit eight copies by regular mail by November 1, 1998. Your submission must include author's name, affiliation, and e-mail address. Authors will be notified of acceptance or rejection by January 20, 1999. The final full version of the accepted paper, for inclusion in the conference proceedings, is due by March 7, 1999.

Contact Address

Gerhard J. Woeginger, Department of Mathematics, TU Graz, Steyrergasse 30, A-8010 Graz AUSTRIA.
Fax: (0043) 316 873 5369;
E-mail: ipco99@opt.math.tu-graz.ac.at;
URL: http://www.opt.math.tu-graz.ac.at/ipco99.

Program Committee

Chair: Gerard P. Cornuejols (Carnegie Mellon University); Rainer E. Burkard (TU Graz); Ravi Kannan (Yale University); Rolf H. Moehring (TU Berlin); Manfred Padberg (New York University); David B. Shmoys (Cornell University); Paolo Toth (University of Bologna); and Gerhard J. Woeginger (TU Graz).

Important Dates

Extended abstracts due: November 1, 1998 (hard copy), November 15, 1998 (electronic); Authors notified: January 20, 1999; Final versions received: March 7, 1999; IPCO '99 Graz: June 9-11, 1999

First Announcement & Call for Papers

Fourth Workshop on Models and Algorithms for Planning and Scheduling Problems
June 14-18, 1999

MAPSP '99

Following three successful workshops at Lake Como, Italy, in 1993, in Wernigerode, Germany, in 1995, and in Cambridge, England, in 1997, the Fourth Workshop on Models and Algorithms for Planning and Scheduling Problems is to be held in Renesse, The Netherlands, June 14-18, 1999. The conference hotel, 'De Zeeuwsche Stromen,' is located in the dunes of Renesse, a beach resort in the province of Zeeland.
The workshop aims to provide a forum for scientific exchange and cooperation in the field of planning, scheduling, and related areas. To maintain the informality of the previous workshops and to encourage discussion and cooperation, there will be a limit of 100 participants and a single stream of presentations.
Contributions on any aspect of scheduling and related fields are welcome.

Conference Organizers

Emile Aarts, Philips Research Laboratories, Eindhoven; Han Hoogeveen, Eindhoven University of Technology; Cor Hurkens, Eindhoven University of Technology; Jan Karel Lenstra, Eindhoven University of Technology; Leen Stougie, Eindhoven University of Technology; and Steef van de Velde, Erasmus University, Rotterdam.

Invited Speakers

Michel Goemans, CORE, Louvain-la-Neuve, Belgium; Martin Grötschel, ZIB, Berlin, Germany; Michael Pinedo, New York University, New York, USA; Lex Schrijver, CWI, Amsterdam, The Netherlands; Eric Taillard, IDSIA, Lugano, Switzerland; Richard Weber, Cambridge University, Cambridge, England; Joel Wein, Polytechnic University, Brooklyn, USA; and Gerhard Woeginger, Technische Universitaet Graz, Austria.

Preregistration

If you are interested in participating, please send an e-mail to mapsp99@win.tue.nl. You will be included in our e-mail list for further notifications. Preregistration does not bear any obligations, but helps us to plan the schedule and keep you informed. In your e-mail please include: last name, first name, affiliation, e-mail address, and whether or not you intend to give a talk.
Presentations will be selected on the basis of a one-page abstract to be submitted no later than March 31, 1999.

Important Dates

July 1, 1998 - Announcement and first call for papers; November 1, 1998 - Second announcement; March 1, 1999 - Deadline for abstract submission; April 1, 1999 - Last date of notification of acceptance; May 1, 1999 - Last date for early registration.
Registration costs include fee and accommodation, based on double room occupancy. Prices mentioned are tentative.
Early registration fee: NLG 800; Late registration fee: NLG 900; Supplement for single room: NLG 125; Beach party: to be announced
The deadline for early registration is May 1, 1999. To register, please consult the conference web site.

Information Sources

For up to date information, consult the conference web site (http://www.win.tue.nl/~mapsp99).

First Announcement

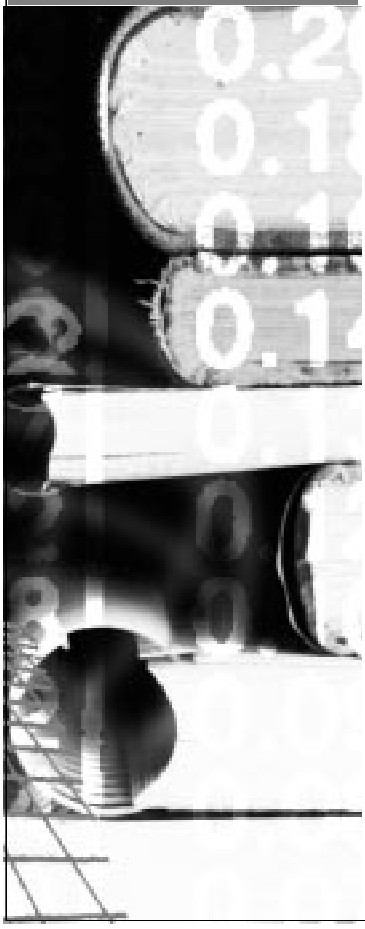
6th Twente Workshop on Graphs and Combinatorial Optimization
26 - 28 May, 1999
University of Twente Enschede, The Netherlands

The Twente Workshop on Graphs and Combinatorial Optimization is organized biennially at the Faculty of Mathematical Sciences at the University of Twente. Topics are: graph theory and discrete algorithms (both deterministic and random) and their applications in operations research and computer science.
We try to keep a 'workshop atmosphere' as much as possible, and so far have succeeded in scheduling no more than two presentations in parallel. We also try to keep the costs as low as possible in order to make the workshop particularly accessible to young researchers.
Prospective speakers are asked to submit an extended abstract of their representation, which will be refereed by a program committee. Your extended abstract should be at least three but not more than four pages and should reach the organizers before March 12, 1999.
The accepted extended abstracts will be collected into a conference volume available at the workshop and published in a volume of Electronical Notes in Discrete Mathematics (ENDM).
The external program committee members include: J.A. Bondy (Lyon); R.H. Möhring (Berlin); R.E. Burkard (Graz); B. Reed (Paris); W.J. Jackson (London); R. Schrader (Cologne); F. Maffioli (Milano); and C. Thomassen (Copenhagen).
A normally refereed special issue of Discrete Applied Mathematics will be devoted to the proceedings of the workshop.

If you are interested in participating in the 6th Twente Workshop, please pre-register now informally. Give your complete postal as well as your e-mail address and indicate whether you would like to give a presentation (ca. 30 min.). If you know the subject and/or title of your presentation, please include that also. You should receive a definite registration form and more detailed information by December 1998.
Further information on the workshop will be available at the web site (http://www.math.utwente.nl/~tw6)

–H.J. BROERSMA, U. FAIGLE, C. HOEDE, J.L. HURINK
Faculty of Mathematical Sciences,
University of Twente, P.O. Box 217,
7500 AE Enschede, The Netherlands
(e-mail: tw6@math.utwente.nl)

Reviews



Addendum to the Book Review from OPTIMA Nº57

Theory and Algorithms for Linear Optimization: An Interior Point Approach

by C. Roos, T. Terlaky and J.-Ph. Vial

Wiley, Chichester, 1997

ISBN 0-471-95676

Even though the book was altogether a pleasure to read, I complained in this review that "the most prominent topic not addressed is infeasible point methods."

I would like to add that while this is true formally, the authors spend a big section of the book on the skew-symmetric model, which is one possibility to avoid having a feasible interior point to start with.

From a practical point of view, it is argued that the slightly larger skew-symmetric model is computationally not significantly more expensive than standard feasible methods, but it enjoys additional theoretical properties.

In view of all this, my above complaint is unjustified, and I would like to add that the book in fact handles the issue of starting points for IP quite elegantly through the extended skew-symmetric model.

-FRANZ RENDL

Gröbner Bases and Convex Polytopes

by Bernd Sturmfels

University Lecture Series Vol. 8, American Mathematical Society

Providence RI, 1995

ISBN 0-8218-0487-1

This book is a state-of-the-art account of the rich interplay between the combinatorics and geometry of convex polytopes and computational commutative algebra, via the tool of Gröbner bases. It is an essential introduction for those who wish to perform research in this fast developing, interdisciplinary field. For the math programmer, this book could be viewed as an exposition of the interactions between integer programming and Gröbner bases.

Gröbner bases of polynomial ideals are special generating sets that depend on certain cost vectors. The discovery of an algorithm for their computation in 1965 by Buchberger catapulted Gröbner bases into a central role in computational commutative algebra and algebraic geometry. (Buchberger named them after his thesis advisor, Wolfgang Gröbner.) An implementation can be found in any of the major computer algebra packages – Macaulay, Reduce, Singular, CoCoA, Maple and Mathematica, to name a few.

The book assumes a working knowledge of the basics of Gröbner bases, polyhedral theory and linear programming. The link between polytopes and algebra is made via a special class of ideals called *toric ideals* which are prime ideals generated by differences of monomials. The material is organized into fourteen chapters, each of which is followed by exercises (some are in fact research projects) as well as brief historical and bibliographic notes. A number of open problems are posed throughout the book, and the reader is quickly brought from basic definitions to the forefront of current research. There is a great deal of emphasis on computational issues, as is evident from the many algorithms and examples included in the book. Many of these computations challenge the ability of current computer algebra packages (and the user's imagination and ingenuity).

The first three chapters treat general polynomial ideals and introduce most of the tools used in this book. The highlights are the notions of *universal Gröbner bases, weight vectors, state polytopes and Gröbner fans.* As seen later, these notions play a crucial role in integer programming. The treatment is custom tailored to the purposes of this book and unique in that it differs from usual presentations of this material. A universal Gröbner basis of an ideal I is a finite subset of the ideal that is a Gröbner basis of I with respect to all weight (cost) vectors. The distinct Gröbner bases of I are in bijection with the vertices of a state polytope of I. The normal fan of a state polytope is the Gröbner fan of I. Chapter 3 presents algorithms for computing state polytopes, Gröbner fans and universal Gröbner bases.

Chapters 4-9 form the heart of the book and are devoted to toric ideals. In Chapter 4, the reader is introduced to toric ideals, their algebraic properties and complexity results for their Gröbner bases. Given a matrix $A \in \mathbb{Z}^{d \times n}$ of rank d the toric ideal of A is the ideal generated by all polynomials (binomials) of the form $x_1^{u_1} x_2^{u_2} \cdot x_n^{u_n} - x_1^{v_1} x_2^{v_2} \cdot x_n^{v_n}$ where $u,v \in \mathbb{N}^n$ and $Au = Av$. A construction of Graver (1975) provides a useful universal Gröbner basis for a toric ideal called the *Graver basis.*

Chapter 5 describes three natural problems that can be associated to the linear map $\pi : \mathbb{N}^n \to \mathbb{N}^d$ such that $x \mapsto A_x$ and its fibers $\pi^{-1}(b)$. (The matrix A is assumed to be in $\mathbb{N}^{d \times n}$ of rank d where $\mathbb{N}$ is the set of non-negative integers.) The first is that of *enumerating* $\pi^{-1}(b) = x \in \mathbb{N}^n : A_x = b$ which amounts to finding all non-negative integer solutions to a system of linear equations. The second is that of randomly generating an element of $\pi^{-1}(b)$, or *sampling*, and the last is to solve the integer program *minimize* $c \cdot x : x \in \pi^{-1}(b)$. It is shown how all three of these problems can be solved using the toric ideal of A and its Gröbner bases. In particular, it is shown that the Gröbner basis of the toric ideal of A with respect to the cost vector c is the unique minimal test set for the family of integer programs *minimize* $c \cdot x : x \in \pi^{-1}(b)$ obtained by varying b.

Chapter 6 treats the case where A has only one row, which amounts to studying knapsack problems. In this case, the elements of the Graver basis are *primitive partition identities.*

The geometry of the universal Gröbner basis of a toric ideal is discussed in Chapter 7. It is shown that the universal Gröbner basis is precisely the set of all edge directions in the convex hulls of the fibers $\pi^{-1}(b)$ as b

varies. The effect of varying the cost function c in the integer programs *minimize* $c \cdot x : x \in \pi^{-1}(b)$ (b varies) while keeping A fixed, is completely captured by the Gröbner fan of the toric ideal of A, making the state polytope of a toric ideal a model for sensitivity in integer programming. Algorithms for computing all of these entities are provided.

Chapter 8 treats the *regular triangulations* of the point configuration $\mathcal{A}$, given by the columns of $\mathcal{A}$. These are simplicial complexes on $\mathcal{A}$ that depend again on cost vectors. The author shows that there is a many-one onto map from the set of all Gröbner bases of the toric ideal of A and the regular triangulations of $\mathcal{A}$. Regular triangulations of $\mathcal{A}$ are in fact the analogs in linear programming of Gröbner bases in integer programming. If Δ_c denotes the regular triangulation of $\mathcal{A}$ induced by the cost vector c, then the maximal simplices in Δ_c are precisely the optimal LP bases of the family of linear programs *minimize* $\{c \cdot x : Ax = b, x \geq 0\}$ as b varies. This approach allows a natural view of integer programming as an arithmetic refinement of linear programming.

Many of the theoretical notions from the previous chapters are illustrated in Chapter 9 using the node-edge incidence matrices of complete graphs (b-matching problems in integer programming).

The last five chapters of the book deal with advanced topics. Chapter 10 generalizes the notion of initial ideals of toric ideals to A-graded algebras. This amounts to *abstract integer programming* where one is allowed to declare a unique random point in each fiber as being optimal as long as the optimal points form an order ideal in $\mathbb{N}^n$. Chapter 11 discusses the role of toric ideals in *canonical subalgebra bases*. Chapter 12 treats certain advanced computational aspects of toric ideals that employ tools from algebra. In particular, localizations of initial ideals (of a toric ideal) are related to group relaxations of integer programs. Toric ideals as defined in this book are related to those found usually in the algebraic geometry literature, in Chapter 13. The book concludes in Chapter 14 with three sophisticated point configurations and properties of their toric ideals.

- REKHA THOMAS

Local Search in Combinatorial Optimization

Edited by Emile Aarts and Jan Karel Lenstra

Wiley, Chichester

ISBN 0-4719-4822-5

"Local Search in Combinatorial Optimization" is the first book I know of that covers under one head many of the interesting aspects of this topic in considerable depth and breadth. At the same time, it finds a fair balance between general theory, methodologies, and applications. The book consists of 13 chapters altogether, each written by leading experts of the respective theme. Fortunately, it is not simply a collection of rather unrelated articles. Due to the editors' editorial experience, their knowledge of the field, their apparent effort in preparing this book, and their careful choice of both the topics and the authors, a rather unique and mostly up-to-date source of theoretical results, different viewpoints, and empirical observations came into being.

Only a formal indication of the interplay between the different chapters is the fair amount of cross references, the common list of references at the end, as well as the joint author and subject indexes. I have followed with great excitement the sometimes different historical perception of different authors and especially the more or less implicit dispute between them, e.g., between the "advertisers" of some methodology like (artificial) neural networks and the potential users of it. This brings me to one of the big pluses of the book at hand. Without any prejudice (but with some humor), the editors gave room for the description and thorough discussion of algorithmic paradigms which caused only some years ago a great irritation between, say, some "pure" combinatorial optimizers on the one side and engineers, practitioners, or scientists from the artificial intelligence community on the other side. In fact, the world of local search has changed dramatically in the last decade and Aarts and Lenstra's book is a tribute to this development. For one thing, incredible changes in computer technology facilitated testing many algorithmic variants and parameter settings on several large problem instances. On the other hand, the development and the recognition of the importance of new algorithmic concepts like simulated annealing, tabu search, and genetic algorithms have changed the landscape significantly. Local search is no longer synonymous with iterative improvement. It is part of the main intention of the editors and authors of this book to present, review, and discuss the current state and the mathematical foundation of these relatively new concepts, as well as their usefulness for solving typical combinatorial problems.

The book is organized in three parts: the complexity of finding locally optimal solutions, algorithmic concepts to compute local optima that are as good as possible, and the application and refinement of local search methods to diverse combinatorial optimization problems. An introductory chapter written by the editors complements the three parts. It gives a first overview of the scene and lays the notational foundation for the rest of the book. However, these suggestions are not always taken up in the subsequent chapters.

The complexity of finding locally optimal solutions is still not known for quite a few combinatorial problems and associated neighborhoods. In response to this fact, Johnson, Papadimitriou, and Yannakakis introduced in 1988 the complexity class PLS and the concept of a PLS-reduction that relates the difficulty of finding local optima between different problems. The second chapter by Mihalis Yannakakis is at the same time a brilliant, pleasant-to-read introduction to, and a rather up-to-date, in-depth survey of the complexity class PLS and PLS-complete problems. In particular, he proves a generic problem to be PLS-complete and presents then several illuminating PLS-reductions to popular problems like the graph partitioning problem under the Kernighan-Lin or the swap neighborhood.

Yannakakis' chapter on the computational complexity of finding a local optimum by any means (not necessarily by a local search algorithm) is followed by a chapter on the worst- and average-case complexity of a certain class of algorithms, written by Craig Tovey. In contrast to the previous chapter, Tovey does not consider specific combinatorial problems and associated neighborhoods, but rather works in the abstract setting of graphs reflecting (data-independent) neighborhood functions. Consequently, the algorithms are assumed to draw information from the neighborhood graph and from an evaluation oracle for the objective function only. In essentially this setting, the following main results are reviewed and proven. For almost all neighborhood functions, any algorithm to find a local optimum must examine at least a constant fraction of the set of all feasible solutions, in the worst case. In the average case, however, even the standard iterative improvement algorithm visits at most a polylogarithmic number of solutions, as long as the degree of the neighborhood graph is sufficiently small. Note, however, that both the lower bound on the worst-case performance as well as the upper bound on the average-case behavior live to a good part from the freedom to choose arbitrary objective functions. For more structured functions, our knowledge remains limited.

At this place it is perhaps suitable to briefly remind the reader of Aarts and Lenstra's book of at least some (uncovered) results that actually complement some of the results, remarks, or questions raised in the first three chapters: the adjacency neighborhood on the 0/1–polytope associated with a linear combinatorial optimization problem (LCOP) is the unique minimal exact neighborhood for local search (Savage 1976); the diameter of any d–dimensional 0/1–polytope is at most d (Naddef 1989); if P is an NP-hard LCOP and N is a neighborhood function such that P, N is in PLS, then N cannot be exact, unless P = NP (Grötschel & Lovász 1996, Schulz, Weismantel & Ziegler 1995); testing the optimality of a given solution is NP-hard, for any NP-hard LCOP.

However, let us come back to the contents of Aarts and Lenstra's book. Each of the Chapters 4 to 7 is devoted to a certain algorithmic paradigm. Chapter 4 is a relatively technical discussion of simulated annealing with a clear emphasis on the stimulating modeling of simulated annealing algorithms by Markov chains. First, the basics of the theory of Markov chains are carefully introduced. Then, the authors elaborate on the results on the probabilistic convergence of simulated annealing algorithms to the set of optimal solutions. Practical issues like the choice of the cooling schedule, the design of parallel algorithms together with the use of neural network models, and the combination of different approaches are covered as well. Alain Hertz, Eric Taillard, and Dominique de Werra survey tabu search in a very concise manner in Chapter 5. The technical details of perhaps the most interesting mathematical result on the probabilistic convergence are only given in the introductory section and the conclusion. The main part of the text focuses on efficiency issues that are illustrated by a few selected examples. In contrast to the two previous chapters, the sixth chapter on genetic algorithms contains a wealth of proofs. In a truly remarkable, unusually personal style, Heinz Mühlenbein responds to other, earlier explanations on why and how genetic algorithms work. His main message is to use mathematical methods that were previously developed to explain phenomena in population genetics. Eventually, we come to perhaps the most controversial technique, artificial neural networks. Carsten Peterson and Bo Söderberg garnish their introduction to this area with quite a few examples and some computational results. In any case, I recommend also reading Section 7 of Chapter 8 which helps to get the results obtained by this method into a better perspective.

Each of the six remaining chapters studies the application of one or more of the previously introduced algorithmic techniques to a specific class of combinatorial optimization problems. David Johnson and Lyle McGeoch investigate the traveling salesman problem; in the broader context of vehicle routing, Michel Gendreau, Gilbert Laporte, and Jean-Yves Potvin also consider all concepts introduced in the second part of the book, whereas extensions of edge exchange neighborhoods are discussed by Gerard Kindervater and Martin Savelsbergh; Edward Anderson, Celia Glass, and Chris Potts review local search algorithms in the wide field of machine scheduling; applications of simulated annealing, tabu search, and genetic algorithms to the different phases in VLSI layout are discussed by Emile Aarts, Peter van Laarhoven, C. Liu, and Peichen Pan; finally, Iiro Honkala and Patric Östergard describe the use of local search methods to design good error-correcting and covering codes. It would be almost unfair to highlight any of these chapters, as all of them are thoroughly and thoughtfully prepared. However, three chapters deserve nevertheless a short special mentioning. First, the chapter on the TSP is a shining example of a carefully designed comparison of the performance of different algorithmic techniques and their implementations. Second, the chapter on machine scheduling captivates by its organization; the authors first extract nicely the common features from the several algorithmic approaches before they actually start considering particular scheduling problems. Third, the really self-contained chapter on VLSI layout is convincing with its scholarly introduction to both the area of layout problems and the local search methods employed to solve them.

I am pleased to report that the 512 pages which make up the book contain only relatively few errors and typos, and only a few of them are annoying (wrong running times, wrong dates, wrong variable names or indices). Another nice and important feature of this book is that although the authors' own contributions have helped to shape the field in the last decades, most chapters are not merely a summary of the authors' own research. Still, some reader might miss the (more detailed) discussion of one or the other quite related topic like Kalai's bound on the diameter of polytopes, Amenta and Ziegler's deformed products, abstract objective functions, or test sets in integer programming. However, as an editor, one has to make a choice.

In summary, this book is a very useful source for researchers and graduate students of quite a range of fields. It gives local search and especially the modern concepts, at which some people still smile, the right (mathematical) standing.

- ANDREAS S. SCHULZ

gallimaufry

The editorial board has been very mobile lately, both in terms of addresses and activities.

- Karen Aardal is spending the fall at the Department of CAAM, Rice University.
- Sebastian Ceria has started a new company called Dash Optimization, Inc., but continues to hold a position at Columbia University.
- Mary Beth Hribar moved to Seattle and is now working for Tera Computer Company. Her new address appears on p.16.
- Robert Weismantel got a position as professor at the University of Magdeburg. His new address appears on p.16 as well.

Deadline for the next issue of OPTIMA is November 30, 1998.

For the electronic version of OPTIMA, please see:
http://www.ise.ufl.edu/~optima/

Application for Membership

I wish to enroll as a member of the Society.

My subscription is for my personal use and not for the benefit of any library or institution.

☐ I will pay my membership dues on receipt of your invoice.

☐ I wish to pay by credit card (Master/Euro or Visa).

CREDITCARD NO. ______________________ EXPIRY DATE ______________________

FAMILY NAME ______________________

MAILING ADDRESS ______________________

____

TELEPHONE NO. ______________________ TELEFAX NO. ______________________

EMAIL ______________________

SIGNATURE ______________________

Mail to:

Mathematical Programming Society
3600 University City Sciences Center
Philadelphia PA 19104-2688 USA

Cheques or money orders should be made payable to The Mathematical Programming Society, Inc. Dues for 1998, including subscription to the journal *Mathematical Programming*, are US $70.
Student applications: Dues are one-half the above rate Have a faculty member verity your student status and send application with dues to above address.

Faculty verifying status

____

Institution

____

O P T I M A

MATHEMATICAL PROGRAMMING SOCIETY

UNIVERSITY OF FLORIDA

Center for Applied Optimization
371 Weil
PO Box 116595
Gainesville FL 32611-6595 USA

FIRST CLASS MAIL

PUBLISHED BY THE
MATHEMATICAL PROGRAMMING SOCIETY &
GATOREngineering® PUBLICATION SERVICES
University of Florida

Journal contents are subject to change by the publisher.