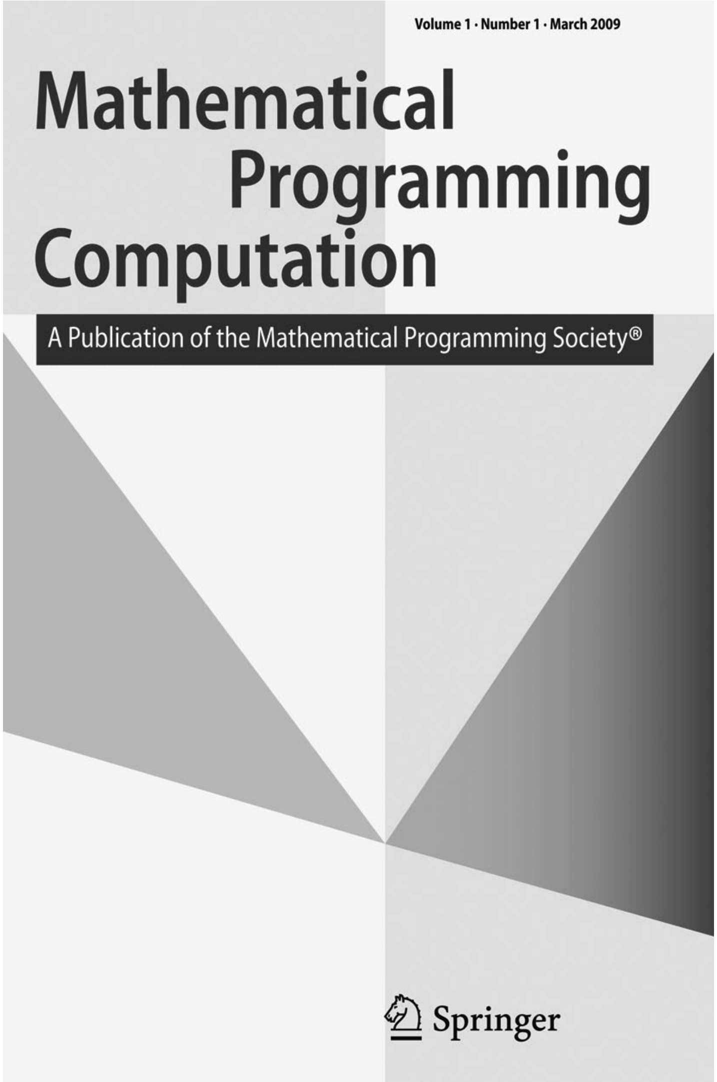
Nº78

OPTIMA

Mathematical Programming Society Newsletter

NOVERMBER 2008



Volume 1 · Number 1 · March 2009

Mathematical Programming Computation

A Publication of the Mathematical Programming Society®

Springer

Mathematical Programming Computation: A New MPS Journal

William Cook, Georgia Institute of Technology
Thorsten Koch, Konrad-Zuse-Zentrum Berlin
September 28, 2008

The Mathematical Programming Society will publish the new journal *Mathematical Programming Computation* (*MPC*) beginning in 2009. The journal is devoted to computational issues in mathematical programming, including innovative software, comparative tests, modeling environments, libraries of data, and/or applications. A main feature of the journal is the inclusion of accompanying software and data with submitted manuscripts. The journal's review process includes the evaluation and testing of the accompanying software. Where possible, the review will aim for verification of reported computational results.

1 Background

In January 2007, Martin Grötschel proposed that MPS consider the creation of a computationally-oriented journal. The proposal was described in an email to Rolf Möhring. The following quote from the email provides a good summary of the intention of the proposal.

> *They see a weakness in our journal landscape concerning information about good codes, the distribution of codes themselves, of data and data collections and everything that has to do with computational aspects of this kind.*

Rolf Möhring formed a committee to explore the idea of a new journal, with members Robert Bixby, William Cook (Chair), Thorsten Koch, Sven Leyffer, David Shmoys, and Stephen Wright. Email discussions were carried out between April 2007 and June 2007, and a short report was sent to Rolf Möhring to wrap up the committee's work. The consensus of the committee was

continues on page 7

78

How to advance in Structural Convex Optimization

Yurii Nesterov

October, 2008

Abstract

In this paper we are trying to analyze the common features of the recent advances in Structural Convex Optimization: polynomial-time interior-point methods, smoothing technique, minimization in relative scale, and minimization of composite functions.

Keywords convex optimization · non-smooth optimization · complexity theory · black-box model · optimal methods · structural optimization · smoothing technique.

Mathematics Subject Classification (2000) 90C06 · 90C22 · 90C25 · 90C60.

Convex Optimization is one of the rare fields of Numerical Analysis, which benefit from existence of well-developed complexity theories. In our domain, this theory was created in the middle of the seventies in a series of papers by A.Nemirovsky and D.Yudin (see [8] for full exposition). It consists of three parts:

– Classification and description of problem instances.
– Lower complexity bounds.
– Optimal methods.

In [8], the complexity of a convex optimization problem was linked with its *level of smoothness* introduced by Hölder conditions on the first derivatives of functional components. It was assumed that the only information the optimization methods can learn about the particular problem instance is the values and derivatives of these components at some test points. This data can be reported by a special unit called *oracle*, and it is *local*, which means that it is not changing if the function is modified far enough from the test point. This model of interaction between the optimization scheme and the problem data is called the *local Black Box*. At the time of its development, this concept fitted very well the existing computational practice, where the interface between the general optimization packages and the problem data was established by Fortran subroutines created independently by the users.

Black-Box framework allows to speak about the lower performance bounds for diffierent problem classes in terms of *informational complexity*. That is the lower estimate for the number of calls of oracle which is necessary for any optimization method in order to guarantee delivering an ε-solution to any problem from the problem class. In this performance measure we do not include at all the complexity of auxiliary computations of the scheme.

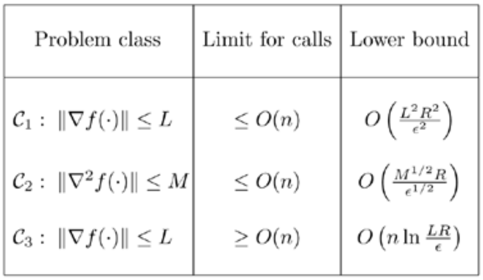
Let us present these bounds for the most important classes of optimization problems posed in the form

$$\min_{x \in Q} f(x), \qquad (1)$$

where $Q \subseteq R^n$ is a bounded closed convex set ($||x|| \leq R$, $x \in Q$), and function f is convex on Q. In the table below, the first column indicates the problem class, the second one gives an upper bound for allowed number of calls of the oracle in the optimization scheme[1], and the last column gives the lower bound for analytical complexity of the problem class, which depends on the absolute accuracy ε and the class parameters.

This paper was written during the visit of the author at IFOR (ETH, Zurich). The author expresses his gratitude to the Scientific Director of this center Hans-Jacob Lüthi for his support and excellent working conditions.

Yurii Nesterov
Center for Operations Research and Econometrics (CORE)
Université catholique de Louvain (UCL)
34 voie du Roman Pays, 1348 Louvain-la-Neuve, Belgium.
Tel.: +32-10-474348
Fax: +32-10-474301
e-mail: Yurii.Nesterov@uclouvain.be



Problem class	Limit for calls	Lower bound
$\mathcal{C}_1 : \ \|\nabla f(\cdot)\| \le L$	$\le O(n)$	$O\left(\frac{L^2R^2}{\epsilon^2}\right)$
$\mathcal{C}_2 : \ \|\nabla^2 f(\cdot)\| \le M$	$\le O(n)$	$O\left(\frac{M^{1/2}R}{\epsilon^{1/2}}\right)$
$\mathcal{C}_3 : \ \|\nabla f(\cdot)\| \le L$	$\ge O(n)$	$O\left(n \ln \frac{LR}{\epsilon}\right)$

(2)

It is important that these bounds are *exact*. This means that there exist methods, which have efficiency estimates on corresponding problem classes proportional to the lower bounds. The corresponding *optimal methods* were developed in [8,9,19,22,23]. For further references, we present a simplified version of the optimal method [9] as applied to the problem (1) with $f \in C_2$:[2]

Choose a starting point $y_0 \in Q$ *and set* $x_{-1} = y_0$. *For* $k \ge 0$ *iterate:*

$$x_k = \arg\min_{x \in Q} \left[f(y_k) + \langle \nabla f(y_k), x - y_k \rangle + \tfrac{M}{2}\|x - y_k\|^2 \right], \qquad (3)$$
$$y_{k+1} = x_k + \tfrac{k}{k+3}(x_k - x_{k-1}).$$

As we see, the complexity of each iteration of this scheme is comparable with that of the simplest gradient method. However, the rate of convergence of method (3) is much faster.

After a certain period of time, it became clear that, despite its mathematical excellence, Complexity Theory of Convex Optimization has a hidden drawback. Indeed, in order to apply convex optimization methods, we need to be sure that functional components of our problem are convex. However, we can check convexity only by analyzing the *structure* of these functions:[3] If our function is obtained from the *basic* convex functions by *convex* operations (summation, maximum, etc.), we conclude that it is convex. If not, then we have to apply general optimization methods which usually do not have theoretical guarantees for the global performance.

Thus, the functional components of the problem are not in the black box the moment we check their convexity and choose minimization scheme. However, we put them into the black box for numerical methods. That is the main conceptual contradiction of the standard Convex Optimization.

Intuitively, we always hope that the structure of the problem can be used for improving the performance of minimization schemes. Unfortunately, structure is a very fuzzy notion, which is quite difficult to formalize. One possible way to describe the structure is to fix the *analytical type* of functional components. For example, we can consider the problems with linear constraints only. It can help, but this approach is very fragile: If we add just a single constraint of another type, then we get a new problem class, and all theory must be redone from scratch.

On the other hand, it is clear that having the structure at hand we can play a lot with the *analytical form* of the problem.We can rewrite the problem in many equivalent settings using non-trivial transformations of variables or constraints, introducing additional variables, etc. However, this would serve almost no purpose without fixing a clear final goal. So, let us try to understand what it could be.

As usual, it is better to look at classical examples. In many situations the sequential reformulations of the initial problem can be seen as a part of numerical scheme. We start from a complicated problem $\mathcal{P}$ and, step by step, change its structure towards to the moment we get a trivial problem (or, a problem which we know how to solve):

$$\mathcal{P} \to \ldots \to (f^*, x^*).$$

A good example of such a strategy is the standard approach for solving system of linear equations

$$Ax = b.$$

We can proceed as follows:

1. Check if A is symmetric and positive definite. Sometimes this is clear from the origin of the matrix.
2. Compute Cholesky factorization of this matrix:
$$A = LL^T\,;$$
where L is a lower-triangular matrix. Form two auxiliary systems
$$Ly = b, \; L^T x = y.$$
3. Solve these system by sequential exclusion of variables.

Imagine for a moment that we do not know how to solve the system of linear equations. In order to discover the above scheme we should apply the following

Golden Rules
1. Find a class of problems which can be solved very efficiently.[a] 2. Describe the transformation rules for converting the initial problem into desired form. 3. Describe the class of problems for which these transformation rules are applicable.
[a] In our example, it is the class of linear systems with triangular matrices.

(4)

In Convex Optimization, these rules were used already several times for breaking down the limitations of Complexity Theory.

Historically, the first example of that type is the theory of polynomial-time interior-point methods (IPM) based on *self-concordant barriers*. In this framework, the class of easy problems is formed by problems of unconstrained minimization of self-concordant functions treated by the Newton method. This *know-how* is further used in the framework of path-following schemes for solving so-called *standard* minimization problems. Finally, it can be shown that by a simple barrier calculus this approach can be extended onto all convex optimization problems with known structure (see [11,18] for details). The efficiency estimates of corresponding schemes are of the order $O(\upsilon^{1/2} \ln \frac{\upsilon}{\varepsilon})$ iterations of the Newton method, where υ is the parameter of corresponding self-concordant barrier. Note that for many important feasible sets this parameter is smaller than the dimension of the space of variables. Hence, for the pure Black-Box schemes such an efficiency is simply unreachable in view of the lower complexity bound for class C_3 (see (2)). It is interesting that formally the modern IPMs look very similar to the usual Black-Box schemes (Newton method plus path-following approach), which were developed in the very early days of Nonlinear Optimization [4]. However, this is just an illusion. For complexity analysis of *polynomial-time* IPM, it is crucial that

[1] If this upper bound is smaller than $O(n)$, then the dimension of the problem is really very big, and we cannot afford the method to perform this amount of calls.

[2] In method (11)-(13) from [9], we can set $a_k = 1 + k/2$ since in the proof we need only to ensure $a_k^2+1 - a_k^2 \le a_k+1$.

[3] Numerical verification of convexity is an extremely difficult problem.

they employ the special barrier functions which *do not satisfy* the local Black-Box assumptions (see [10] for discussion).

The second example of using the rules (4) needs more explanations. By certain circumstances, these results were discovered with a delay of twenty years. Perhaps they were too simple. Or maybe they are in a seemingly very sharp contradiction with the rigorously proved lower bounds of Complexity Theory.

Anyway, now everything looks almost evident. Indeed, in accordance to Rule 1 in (4), we need to find a class of very easy problems. And this class can be discovered *directly* in Table (2)! To see that, let us compare the complexity of the classes $C1$ and $C2$ for the accuracy of 1% ($\varepsilon = 10^{-2}$). Note that in this case, the accuracy-dependent factors in the efficiency estimates vary from ten to ten thousands. So, the natural question is:

> *Can the* easy problems *from C_2 help us somehow in finding an approximate solution to the* difficult problems *from C_1?*

And the evident answer is: Yes, of course! It is a simple exercise in Calculus to show that we can always approximate a Lipschitz-continuous nonsmooth convex function on a bounded convex set with a uniform accuracy $\varepsilon > 0$ by a smooth convex function with Lipschitz-continuous gradient. We pay for the accuracy of approximation by a large Lipschitz constant M for the gradient, which should be of the order $O(\frac{1}{\epsilon})$. Putting this bound for M in the efficiency estimate of C_2 in (2), we can see that in principle, it is possible to minimize nonsmooth convex functions by the oracle-based gradient methods with analytical complexity $O(\frac{1}{\epsilon})$. But what about the Complexity Theory? It seems that it was *proved* that such efficiency is just impossible.

It is interesting that in fact we do not get any contradiction. Indeed, in order to minimize a smooth approximation of nonsmooth function by an oracle-based scheme, we need to change the initial oracle. Therefore, from mathematical point of view, we violate the Black-Box assumption. On the other hand, in the majority of practical applications this change is not difficult. Usually we can work directly with the structure of our problem, at least in the cases when it is created by us.

Thus, the basis of the *smoothing technique* [12,13] is formed by two ingredients: the above observation, and a trivial but systematic way for approximating a nonsmooth function by a smooth one. This can be done for convex functions represented explicitly in a max-form:

$$f(x) = \max_{u \in Q_d} \{ \langle Ax - b, u \rangle - \phi(u),$$

where Q_d is a bounded and convex dual feasible set and $\phi(u)$ is a concave function. Then, choosing a nonnegative strongly convex function $d(u)$, we can define a smooth function

$$f_\mu(x) = \max_{u \in Q_d} \{ \langle Ax - b, u \rangle - \phi(u) - \mu \cdot d(u)\} \qquad (5)$$

which approximates the initial objective. Indeed, denoting $D_d = \max\limits_{u \in Q_d} d(u)$,

we get

$$f(x) \ge f_\mu(x) \ge f(x)\ \ \mu D_d.$$

At the same time, the gradient of function f_μ is Lipschitz-continuous with Lipschitz constant of the order of $O(\frac{1}{\mu})$ (see [12]) for details).

Thus, we can see that for an *implementable* definition (5), we get a possibility to solve problem (1) in $O(\frac{1}{\epsilon})$ iterations of the fast gradient method (3). In order to see the magnitude of the improvement, let us look at the following example:

$$\min_{x \in \Delta_n} \left[f(x) \stackrel{\mathrm{def}}{=} \max_{1 \le j \le m} \langle a_j, x \rangle \right], \qquad (6)$$

where $\Delta_n \in R^n$ is a standard simplex. Then the properly implemented smoothing technique ensures the following rate of convergence:

$$f(x_N) - f^* \le \frac{4\sqrt{\ln n \cdot \ln m}}{N} \cdot \max_{i,j} |a_j^{(i)}|.$$

If we apply to problem (6) the standard subgradient methods (e.g. [14]), we can guarantee only

$$f(x_N) - f^* \le \frac{\sqrt{\ln n}}{\sqrt{N+1}} \cdot \max_{i,j} |a_j^{(i)}|.$$

Thus, up to a logarithmic factor, for obtaining the same accuracy, the methods based on smoothing technique need only a square root of iterations of the usual subgradient scheme. Taking into account, that usually the subgradient methods are allowed to run many thousands or even millions of iterations, the gain of the smoothing technique in computational time can be enormously big.[4]

It is interesting, that for problem (6) the computation of the smooth approximation is very cheap. Indeed, let us use for smoothing the *entropy function*:

$$d(u) = \ln m + \sum_{i=1}^{n} u^{(i)} \ln u^{(i)}, \quad u \in \Delta_m.$$

Then the smooth approximation (5) of the objective function in (6) has the following compact representation:

$$f_\mu(x) = \mu \ln \left[\frac{1}{m} \sum_{j=1}^{m} e^{\langle a_j, x \rangle / \mu} \right].$$

Thus, the complexity of the oracle for $f(x)$ and $f_\mu(x)$ is similar. Note that again, as in the polynomial-time IPM theory, we apply the standard oracle-based method ((3) in this case) to a function which does not satisfy the Black-Box assumptions.

An inexplicable blindness to the possibility to reduce the complexity of nonsmooth optimization problems with known structure is not restricted to the smoothing technique only. As it was shown in [7], very similar results can be obtained by the extra-gradient method by G. Korpelevich [6] using the fact that this method is optimal for the class of variational inequalities with Lipschitz-continuous operator (for these problems it converges as $O(\frac{1}{k})$). Actually, in a verbal form, the optimality of the extra-gradient method was known already for a couple of decades. However, a rigorous proof of this important fact and discussion of its consequences for Structural Nonsmooth Optimization was published only in [7], after discovering the smoothing technique.

To conclude this section, let us discuss the last example of acceleration strategies in Structural Optimization. Consider the problem of minimizing the *composite*

[4] It is easy to see that the standard subgradient methods for nonsmooth convex minimization need indeed $O(\frac{1}{\epsilon^2})$ operations to converge. Consider a univariate function $f(x) = |x|$, $x \in R$. Let us look at the subgradient process:

$$x_{k+1} = x_k - h_k f'(x_k), \quad x_0 = 1, \quad h_k = \frac{1}{\sqrt{k+1}} + \frac{1}{\sqrt{k+2}}, \quad k \ge 0.$$

It easy to see that $|x_k| = \frac{1}{\sqrt{k+1}}$. However, the step-size sequence is optimal [8].

objective function:

$$\min_{x \in R^n} \; [\, f(x) + \Psi(x) \,], \qquad (7)$$

where the function *f* is a convex diffierentiable function on dom Ψ with Lipschitz-continuous gradient, and function Ψ is an *arbitrary* closed convex function. Since Ψ can be even discontinuous, in general this problem is very difficult. However, if we assume that function Ψ is *simple*, then the situation is changing. Indeed, suppose that for any $\bar{y} \in$ dom Ψ we are able to solve explicitly the following auxiliary optimization problem:

$$\min_{x \in \mathrm{dom}\,\Psi} \Big[f(\bar{y}) + \langle \nabla f(\bar{y}), x - \bar{y} \rangle + \frac{M}{2} \|x - \bar{y}\|^2 + \Psi(x) \Big] \qquad (8)$$

(compare with (3)). Then it becomes possible to develop for problem (7) fast gradient methods (similar to (3)), which have the rate of convergence of the order $O(\frac{1}{k^2})$ (see [15] for details; similar technique was developed in [3]). Note that the formulation (7) can be also seen as a part of Structural Optimization since we use the knowledge of the structure of its objective function directly in the optimization methods.

Conclusion

In this paper, we have considered several examples of significant acceleration of the usual oracle-based methods. Note that the achieved progress is visible only because of the supporting complexity analysis. It is interesting that all these methods have some prototypes proposed much earlier:

– Optimal method (3) is very similar to the *heavy point* method:

$$x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1}),$$

where α and β are some fixed positive coefficients (see [20] for historical details).

– Polynomial-time IPM are very similar to *some* variants of the classical barrier methods [4].

– The idea to apply smoothing for solving minimax problems is also not new (see [21] and the references therein).

At certain moments of time, these ideas were quite new and attractive. However, they did not result in a significant change in computational practice since they were not provided with a convincing complexity analysis. Indeed, many other schemes have similar theoretical justifications and it was not clear at all why these particular suggestions deserve more attention. Moreover, even now, when we know that the modified variants of some old methods give excellent complexity results, we cannot say too much about the theoretical efficiency of the original schem(7)es.

Thus, we have seen that in Convex Optimization the complexity analysis plays an important role in selecting the promising optimization methods among hundreds of others. Of course, it is based on investigation of the worst-case situation. However, even this limited help is important for choosing the perspective directions for further research. This is true especially now, when the development of Structural Optimization makes the problem settings and corresponding efficiency estimates more and more interesting and diverse.

The size of this paper does not allow us to discuss other interesting setting of Structural Convex Optimization (e.g. optimization in relative scale [16, 17]). However, we hope that even the presented examples can help the reader to find new and interesting research directions in this promising field (see, for example, [1,2,5]).

References

1. A. d'Aspremont, O. Banerjee, and L. El Ghaoui. First-Order Methods for Sparse Covariance Selection. *SIAM Journal on Matrix Analysis and its Applications*, 30(1), 56-66, (2008).
2. O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model Selection Through Sparse Maximum Likelihood Estimation. *Journal of Machine Learning Research*, 9, 485-516 (2008).
3. A. Beck and M. Teboulle. A Fast Iterative Shrinkage-Threshold Algorithm Linear Inverse Problems. Research Report, Technion (2008).
4. A.V. Fiacco and G.P. McCormick. Nonlinear Programming: Sequential Uncon strained Minimization Technique. *John Wiley*, New York, 1968.
5. S. Hoda, A. Gilpin, and J. Pena. Smoothing techniques for computing Nash equilibria of sequential games. Research Report. Carnegie Mellon University, (2008).
6. G.M. Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon* 12, 747-756 (1976).
7. A. Nemirovski. Prox-method with rate of convergence *O*(1/*t*) for variational inequalities with Lipschitz continuous monotone operators and smooth convex concave saddle point problems. *SIAM Journal on Optimization*, 15, 229-251 (2004).
8. A. Nemirovsky and D. Yudin. Problem Complexity and Method Efficiency in Optimization. *Wiley*, New-York, 1983.
9. Yu. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O\left(\frac{1}{k^2}\right)$. *Doklady AN SSSR* (translated as Soviet Mathematics Doklady), 269(3), 543-547 (1983).
10. Yu. Nesterov. Interior-point methods: An old and new approach to nonlinear progamming. *Mathematical Programming*, 79(1-3), 285-297 (1997).
11. Yu. Nesterov. Introductory Lectures on Convex Optimization. *Kluwer*, Boston, 2004.
12. Yu. Nesterov. Smooth minimization of non-smooth functions. CORE Discussion Paper 2003/12 (2003). Published in *Mathematical Programming*, 103 (1), 127-152 (2005).
13. Yu. Nesterov. Excessive gap technique in nonsmooth convex minimization. *SIAM Journal on Optimization,* 16 (1), 235-249 (2005).
14. Yu. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming* (2007)
15. Yu. Nesterov. Gradient methods for minimizing composite objective function. *CORE Discussion Paper* 2007/76, (2007).
16. Yu. Nesterov. Rounding of convex sets and efficient gradient methods for linear programming problems. *Optimization Methods and Software*, 23(1), 109-135 (2008).
17. Yu. Nesterov. Unconstrained convex minimization in relative scale. Accepted by *Mathematics of Operations Research*.
18. Yu. Nesterov, A. Nemirovskii. Interior point polynomial methods in convex programming: Theory and Applications. *SIAM*, Philadelphia, 1994.
19. B.T. Polyak. A general method of solving extremum problems. *Soviet Mat. Dokl.* 8, 593-597 (1967)
20. B. Polyak. Introduction to Optimization. *Optimization Software*, New York, 1987.
21. R. Polyak. Smooth Optimization Methods for Minimax Problems. *SIAM J. Control and Optimization*, 26(6), 1274-1286 (1988).
22. N.Z. Shor. Minimization Methods for Nondiffierentiable Functions. *Springer-Verlag*, Berlin, 1985.
23. S.P. Tarasov, L.G. Khachiyan, and I.I. Erlikh. The Method of Inscribed Ellipsoids. *Soviet Mathematics Doklady*, 37, 226– 230 (1988).

MPS Chair's Column – Optima 78

Steve Wright
15 October 2008

We note with sadness the passing of Garth McCormick, who died in Maryland on August 24. Professor McCormick's 1968 book with Anthony Fiacco on logarithmic barrier methods for nonlinear programming was acclaimed at the time of its publication, then had a second life during the interior-point revolution starting in the mid-1980s, when it was recognized for its seminal contributions. He was also a pioneer in computational differentiation, and worked on many applications of nonlinear programming. We send our deepest condolences to his family.

I'm delighted with the launch of the new MPS journal Mathematical Programming Computation, whose first issue will appear in 2009. An article by editor-in-chief Bill Cook and general editor Thorsten Koch describing the genesis of the journal, its aims and scope, and the large and distinguished editorial staff, appears in this issue of Optima. MPC (as we inevitably refer to it) is innovative in several respects, including its strong focus on software and computation, its mechanism for evaluating contributions, and its means of distribution (freely available online, with print edition published by Springer and included in MPS membership). My thanks to all who serve as editors and advisors, and especially to Bill and Thorsten for their tireless work in booting up the journal. Now it is up to us to do some great computational research and send our papers (and software) to MPC!

Planning for our Society's flagship event, ISMP in Chicago (August 23-29, 2009), continues to pick up speed, as you can see from the web site www.ismp2009.org . The plenary and semi-plenary speakers have been announced, as has the list of clusters and cluster organizers. Registration and abstract submission through the web site will be available in November, along with hotel information. Please contact a cluster organizer in the relevant area if you wish to speak or organize a session.

Prizes sponsored by MPS and fellow societies will be awarded at the opening ceremony of ISMP, at Orchestra Hall in Chicago. These are the leading prizes in our discipline, and I urge you to visit the Society's web site www.mathprog.org for information on prizes and the current calls for nominations, and think about nominating your most deserving colleagues for these honors.

To those members who are not yet regular users of Optimization Online (www.optimization-online.org) I urge you to take a look at this valuable service. It contains a repository of optimization preprints, which you can browse and contribute to. You can also sign up for a digest that is emailed at the start of each month. The cgi scripts underlying the site have held up well without extensive modification since being written by Jean-Pierre Goux in 2000, though there is the occasional hiccup because of server transitions or security problems at Argonne, the host site. In the coming months, we hope to find the time, resources, and expertise to improve the utility of the site for optimization researchers and users.

There are a number of Society initiatives in the pipeline that you'll hear more about in future columns. We have been working on a new web site, which will be brought online at the current URL in the coming months. The new site will be easier to maintain and will scale better as the amount of content grows. We're also working on revisions to the Society's constitution and by-laws, modernizing them and bringing them into line with current practice, and into conformity with the standards expected for nonprofit organizations. The revisions to the bylaws have been quite extensive, with the Society's council and executive committee deeply involved in the process. We will be presenting the modified constitution to members in coming months, with a view to ratifying it at the Society's business meeting at ISMP 2009.

You should by now have received membership renewal notices for your next year membership in MPS. 2009 will be a banner year for the Society, because of ISMP, the new journal, and the other initiatives mentioned above and in your renewal letter. I look forward with keen anticipation to your continued participation, especially those members who joined as a result of their attendance at ICCOPT 2007.

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).

CREDIT CARD NO. EXPIRATION DATE

FAMILY NAME

MAILING ADDRESS

TELEPHONE NO. TELEFAX NO.

E-MAIL

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 2008, including subscription to the journal *Mathematical Programming*, are US $85. Retired are $40. Student applications: Dues are $20. Have a faculty member verify your student status and send application with dues to above address.

Faculty verifying status

Institution

continues from page 1

to recommend that MPS possibly move forward with a web-based journal.

An MPC proposal was delivered to the MPS Council in September and approved on November 11, 2007. Following this, negotiations began with Springer Verlag concerning possible distribution of the journal.

On July 9, 2008, the MPS Council unanimously approved the following two motions.

> *1. Council approves the establishment of Mathematical Programming Computation (MPC) as a journal of the Society, following the guidelines proposed in the attached document "Mathematical Programming Computation: Notes on a New MPS Journal", with William Cook as the first editor-in-chief and Thorsten Koch as the first general editor. Council further approves the initial advisory board listed in this document.*
>
> *2. Council approves the proposed contract with Springer-Verlag GmbH attached to this message, concerning the publication of Mathematical Programming Computation.*

And MPC was off and running! The formation of the MPC Editorial Board was completed in August 2008 and the first manuscript was submitted to MPC on September 9, 2008.

The directors of the INFORMS Optimization Society and the SIAM Activity Group on Optimization have been contacted regarding MPC. Both organizations strongly support the plans for the journal. Discussions with the COIN-OR Technical Board have taken place over the past year, focusing on possible connections between MPC and the COIN-OR services.

2 Journal Distribution

MPC will be published together with Springer Verlag, with the first volume, consisting of four issues, appearing in 2009. The partnership with Springer creates an attractive combination of accessibility for both authors and academic institutions.

All MPS members will receive print versions of the journal as part of their membership benefits. The contents of the journal will be made freely available on the society-run MPC web site mpc.zib.de, housed at the Konrad-Zuse-Zentrum Berlin (ZIB). Supplementary material will be included on the web site, supporting the computational studies described in the journal articles.

3 Aims and Scope

MPC publishes original research articles concerning computational issues in mathematical programming. Topics covered in MPC include linear programming, convex optimization, nonlinear optimization, stochastic optimization, robust optimization, integer programming, combinatorial optimization, global optimization, network algorithms, and modeling languages.

MPC supports the creation and distribution of software and data that foster further computational research. The opinion of the reviewers concerning this aspect of the provided material is a considerable factor in the editorial decision process. Another factor is the extent to which the reviewers are able to verify the reported computational results. To these aims, authors are highly encouraged to provide the source code of their software. Submitted software is archived with the corresponding research articles. The software is not updated and the journal is not intended to be the point of distribution for the software. The author's licensing information is included with the archived software. In case the software is no longer available through other means, MPC will distribute it on individual request under the license given by the author. Our intent is to at least partly remedy today's situation where it is often impossible to compare new results with those computed by other codes several years ago.

Articles describing software where no source code is made available are acceptable, provided reviewers are given access to executable codes that can be used to evaluate reported computational results. Articles may also provide data, their description, and analysis. Articles not providing any software or data will be considered, provided they advance the state-of-the-art regarding a computational topic.

4 Information for Authors

Manuscript

Only articles written in the English language will be considered for publication. There is no pre-set page limit on articles, but the journal encourages authors to be concise. The length of the manuscript will be taken into consideration in the review process. Authors should aim to present summaries of computational tests, rather than long tables of individual results. Detailed tables and log files can be included in supplementary material to be made available on the journal's Web site.

Articles should give a general description of the software, its scope, and the algorithms used. Rather than long presentations of well-known algorithms, authors are encouraged to give details that deviate from the known state-of-the-art on specific design decisions and their consequences and implementation details.

Software

Computer codes must be accompanied by a clear description of the environment in which they are expected to be built, including instructions on how to obtain any required third-party packages. Clear and easy to follow instructions must be given on how to build and run the author's software, and how to use it to recompute any computational results given in the article.

Submission

Authors are invited to submit articles for possible publication in MPC. Articles can be submitted in Adobe PDF format through the journal's web-based system at mpc.zib.de. Software and supplementary material can also be submitted through this system. Software should be delivered as a zip or gzipped-tar archive file that unpacks into a directory, reflecting the name of the software.

Review

Articles within the scope of the journal will receive a rigorous review. The editorial board will strive to have papers reviewed within a four-month period. This target will be extended in cases of exceptionally long or difficult manuscripts.

The review of articles describing software will include an evaluation of the computer codes received with the submitted manuscript. The criteria used in the software review include the following points.

1. The innovation, breadth, and depth of the contribution.
2. An evaluation of the progress in performance and features compared with existing software.
3. The conditions under which the software is available.
4. The availability and quality of user documentation.
5. The accessibility of the computer code; the ease with which a developer can make modifications.

5 Editorial Board

The structure of the MPC Editorial Board is similar to that of the Mathematical Programming Series A board, with an additional team of Technical Editors to carry out software evaluations. It has been suggested that MPC adopt the flat model used in SIAM journals, with the aim of reducing average review times. Although we are not adopting this SIAM-like structure, this point can be revisited if a significant percentage of review times are above the four-month target.

Editor-in-Chief

The Editor-in-Chief has the overall responsibility for the journal. The duties include the formation of the Editorial Board, the establishment of guidelines and quality standards for the review process, oversight of the timeliness and fairness of reviews, the assignment of manuscripts to Area Editors, light copy editing of final manuscripts, and the general promotion of the journal. The initial Editor-in-Chief is William Cook (Georgia Tech).

General Editor

The General Editor is responsible for the quality of the software evaluation. The duties include consulting with the Editor-in-Chief on the selection of a board of Technical Editors, providing guidelines to the TE board, serving as a contact with the hardware/software support group, and assisting in setting up testing facilities. The initial General Editor is Thorsten Koch (ZIB).

Production Editor

The Production Editor is responsible for building and maintaining the MPS web distribution of the journal, including an on-line submission process. The initial Production Editor is Wolfgang Dalitz (ZIB).

Advisory Board

An Advisory Board provides general oversight of the journal. Membership on the board is subject to approval by the MPS Publications Committee and the MPS Council. The initial board consists of the following members.

- Robert Bixby (Rice University)
- Donald Goldfarb (Columbia University)
- Nick Gould (Rutherford Appleton Laboratory)
- Martin Grötschel (Konrad-Zuse-Zentrum Berlin)
- David Johnson (AT&T Research)
- Kurt Mehlhorn (Max-Planck-Institut Saarbrücken)
- Hans Mittelmann (Arizona State University)
- Arkadi Nemirovski (Georgia Tech)
- Jorge Nocedal (Northwestern University)
- Michael Trick (Carnegie Mellon University)
- Robert Vanderbei (Princeton University)
- David Williamson (Cornell University)

Area Editors

Area Editors have direct contact with authors, carry out initial reviews of papers, make assignments to Associate Editors and to Technical Editors, and make editorial decisions to accept or decline submissions. The initial board of Area Editors and their areas of interest are given in the following list.

- Daniel Bienstock (Columbia University): Linear and Integer Programming
- Robert Fourer (Northwestern University): Modeling Languages and Systems
- Andrew V. Goldberg (Microsoft Research): Graph Algorithms and Data Structures
- Sven Leyffer (Argonne National Laboratory): Nonlinear Optimization
- Jeffrey T. Linderoth (Univ. of Wisconsin-Madison): Stochastic Optimization, Robust Optimization, and Global Optimization
- Gerhard Reinelt (University of Heildelberg): Combinatorial Optimization
- Kim-Chuan Toh (National University of Singapore): Convex Optimization

Associate Editors

Papers are assigned to an Associate Editor by one of the Area Editors or by the Editor-in-Chief. The AE will seek to obtain reviews from at least two referees (one of whom could be the AE), or in the case of weaker papers a single negative report. The identity of the assigned AE is not revealed to the author of the paper. The members of the AE board are as follows.

- Shabbir Ahmed, Georgia Tech
- Samuel Burer, University of Iowa
- Alberto Caprara, University of Bologna
- Sanjeeb Dash, IBM TJ Watson Research Center
- Camil Demetrescu, University of Rome
- Matteo Fischetti, University of Padova
- Emmanuel Fragniere, HEG, Geneva
- Michael P. Friedlander, University of British Columbia
- Jacek Gondzio, University of Edinburgh
- Philip E. Gill, University of California San Diego
- Oktay Günlük, IBM TJ Watson Research Center
- Michal Kocvara, University of Birmingham
- Adam N. Letchford, Lancaster University
- Andrea Lodi, University of Bologna
- François Margot, Carnegie Mellon University
- Rafael Marti, University of Valencia
- Laurent Michel, University of Connecticut
- David Pisinger, University of Copenhagen
- Nikolaos V. Sahinidis, Carnegie Mellon University
- Peter Sanders, University of Karlsruhe
- Melvyn Sim, National University of Singapore
- Huseyin Topaloglu, Cornell University
- Michael Ulbrich, Technische Universität München
- Andreas Wächter, IBM TJ Watson Research Center
- Renato Werneck, Microsoft Research
- Yin Zhang, Rice University

continue on page 11

Discussion Column

Alexandre d'Aspremont
Javier Peña
Katya Scheinberg

The smoothing method which Yurii Nesterov describes in his article turns out not only to be an important theoretical advance in the first order method, but also an inspiration for new approaches for many large scale convex optimization problems.

We present two articles discussing the use of smooth first order methods for two different settings, for which classical approaches such as interior point methods failed to produce efficient methods. One is a classical convex nonsmooth setting - semidefinite programming. While semidefinite programs are inherently non-smooth, the smoothing and projection subproblems formed in the smoothing argument detailed in the previous article can be solved explicitly for a wide class of semidefinite optimization problems. This means that smooth first-order methods offer an alternative to interior point methods for solving large-scale semidefinite programs.

The other setting is computation of Nash equilibria of large sequential two-person, zero-sum games with imperfect information. The smoothing approach has enabled the authors to find near-equilibria for a four-round model of Texas Hold'em poker — a problem that is several orders of magnitude larger than what was previously computable. A poker player based on the resulting strategies turns out to be superior to most automatic poker players and is competitive with the best human poker players.

Smooth Semidefinite Optimization

Alexandre d'Aspremont

1. Introduction

Semidefinite programming has received a significant amount of attention since [NN94] extended the classic complexity analysis of Newton's method to a much broader class of functions. Since then, and despite their somewhat abstract nature, semidefinite programs have found a wide array of applications in engineering (see [BV04]), combinatorial optimization (see [Ali95,GW95]), statistics (see [BEGd07]) or machine learning (see [SBXD05,WS06,LCB+02]) for example.

A number of efficient numerical packages have been developed to solve relatively large semidefinite programs using interior point algorithms based on Newton's method (see [Mit03] for an early survey). These codes exhibit both rapid convergence and excellent reliability. In particular, the quadratic convergence of Netwon's method near the optimum mean that high precision solutions can be obtained by running only a few more iterations. However, because these methods form second order (Hessian) information to compute the Newton step, they have very high memory requirements, as well as a very high numerical cost per iteration.

At the other end of the complexity spectrum lie bundle type methods (see [HR00,Ous00] for example) which directly apply nonsmooth to semidefinite optimization problems after an appropriate choice of subdifferential. These methods only require forming a subgradient at each iteration, which in practice means computing a few leading eigenvalues, hence have a very low computational/memory cost per iteration. However, early bundle algorithms had a dependence in the precision target $\varepsilon > 0$ of $O(1/\varepsilon^2)$ which restricted their use to applications with a very coarse precision target. While some early first order algorithms produced improved bounds of $O(1/\varepsilon)$, they remained relatively specialized. Furthermore, bundle methods required a significant number of input parameters to be manually tuned to pick an appropriate subgradient and improve efficiency.

The smoothing argument developed in [Nes05] and discussed in the previous paper solves both these issues at once. It proves convergence of smooth optimization methods on a wide class of semidefinite optimization problems with an explicit bound of $O(1/\varepsilon)$ on the total number of iterations, and since the value of most parameters are explicit functions of the problem data, these methods require no parameter tuning.

In what follows, we show how the smooth optimization algorithm detailed in [Nes05] was used in [dEGJL07] to solve a sparse PCA problem. We refer the reader to [Nes07], [Ous00], [Nem04] or [LNM07] for further details and alternative algorithms with similar characteristics.

2. Semidefinite Optimization

For simplicity here, we will focus on the particular semidefinite program formed in [dEGJL07] to bound sparse eigenvalues of covariance matrices. Given a symmetric matrix $A \in \mathbf{S}_n$, we seek to solve:

$$\begin{array}{ll} \text{minimize} & \lambda_{\max}(A+U) \\ \text{subject to} & |U_{ij}| \leq \rho, \end{array} \qquad (1)$$

in the variable $U \in \mathbf{S}_n$. The objective of this problem is not smooth: when the leading eigenvalue $\lambda_{\max}(A+U)$ is not simple, the function is non-differentiable (see [Lew03,OW95] for details and an explicit derivation of the Hessian). The smoothing argument in [Nes05] first finds a smooth approximation of the objective function in (1) then applies an optimal first-order

Acknowledgements. The author would like to acknowledge support from NSF grant DMS-0625352, NSF CDI grant SES-0835550, ONR grant number N00014-07-1-0150, a Howard B. Wentz junior faculty award and a Peek junior faculty fellowship.

Alexandre d'Aspremont: ORFE, Princeton University, Princeton NJ 08544, USA.
e-mail: aspremon@princeton.edu

Mathematics Subject Classification (1991): 44A12, 44A60, 90C05, 90C34, 91B28
Correspondence to: Alexandre d'Aspremont

method to the smooth problem. The benefit of the smoothing step is that optimizing (1) using nonsmooth method has a complexity of $O(1/\varepsilon^2)$ while we will see that the smooth approximate problem can be solved with a complexity of $O(1/\varepsilon)$.

2.1. Smoothing

In this case, smoothing the objective in (1) turns out to be relatively straight-forward. Indeed, the function

$$f_\mu(X) = \mu \log (\mathbf{Tr} \exp(X/\mu))$$

is a uniform ε-approximation of $\lambda \max(X)$. Furthermore, it was shown in [Nes07] or [Nem04] that its gradient $\nabla f_\mu(X)$ is Lipschitz continuous with constant

$$L = \frac{1}{\mu} = \frac{\log n}{\epsilon}$$

The tradeoff between the quality of the approximation (controlled by ε) and the smoothness of the gradient (given by $L = \log n/\varepsilon$) is here completely explicit. We will now apply a smooth minimization algorithm to the smooth approximation of problem (1).

2.2. Smooth minimization

We now apply a slightly more elaborate version of the optimal first-order algorithm detailed in equation (3) in the previous paper. Let us write

$$Q_1 = \{U \in S_n : |U_{ij}| \leq \rho\}$$

the algorithm proceeds as follows.

```
Repeat:
1. Compute $f_\mu(A + U_k)$ and $\nabla f_\mu(A + U_k)$
2. Find $Y_k = \arg\min_{Y \in Q_1}$
      $\langle \nabla f_\mu(U_k), Y \rangle + \frac{1}{2} L \|U_k - Y\|_F^2$
3. Find $W_k = \arg\min_{W \in Q_1}$
   $\left\{ \frac{L\|W\|^2}{2} + \sum_{i=0}^{k} \frac{i+1}{2} (f_\mu(U_i)\right.$
   $\left. + \langle \nabla f_\mu(U_i), W - U_i \rangle) \right\}$
4. Set $U_{k+1} = \frac{2}{k+3} W_k + \frac{k+1}{k+3} Y_k$
Until gap ≤ ε.
```

Step 1 is the most computationally intensive step in the algorithm and involves computing a matrix exponential to derive the gradient of $f_\mu(A+U)$. Steps 2 and 3 are simply Euclidean projections on the unit box in $\mathbf{S}_n$.

2.3. Implementation

To compute the gradient $\nabla f_\mu(A + U)$ without overflows, we can evaluate it as:

$$\nabla f_\mu(A+U) = \frac{\exp((A+U-\lambda\mathbf{I})/\mu)}{\mathbf{Tr}\,(\exp((A+U-\lambda\mathbf{I})/\mu))}$$

having set $\lambda = \lambda_{\max}(A + U)$. Computing the gradient thus means computing a matrix exponential, which is a classic linear problem (see [MVL03] for a complete discussion) and has a complexity of $O(n^3)$.

In fact this expression for the gradient also provides us with an intuitive interpretation of the connection between the smoothing technique and bundle methods. Because the matrix exponential can also be written:

$$\sum_{i=1,\ldots,n} \exp(\lambda_i(A+U-\lambda\mathbf{I})/\mu)\mathbf{v_i}\mathbf{v_i^T}$$

where λ_i and v_i are the eigenvalues and eigenvectors of the matrix $(A + U - \lambda I)/\mu$ in decreasing order, the gradient can be seen as a bundle of subdifferentials $v_i v_i^T$ with weights decreasing exponentially with λ_i. When the objective function is smooth $\lambda_{\max}(A + U)$ is well separated from the rest of the spectrum and the smooth gradient is close to the nonsmooth one $v_1 v_1^T$, when the leading eigenvalues are tightly clustered however the gradient will be a mixture of subdifferentials. In that sense, the smooth semidefinite minimization algorithm can be seen as a bundle method whose weights are adjusted adaptively with smoothness. Pushing the argument a bit further, one can show that only a few eigen-values are often required to approximate with a precision sufficient to maintain convergence (see [d'A05] for details).

2.4. Other examples

The success of the smoothing technique in the previous examples stems from the fact that smooth uniform approximations could be computed efficiently (analytically in fact) and that both projections on the feasible set were available in closed form. This situation is far from being exceptional and the smoothing technique has been applied to several other semidefinite optimization problems which require solving large-scale dense instances with relatively low precision. In fact, projecting on simple feasible sets is often much simpler than forming the corresponding barrier or computing a Newton step.

Another problem instance where smooth optimization methods have proved very efficient is covariance selection (see [dBEG06]). Here we solve,

$$\min_{\{X \in \mathbf{S}_n\ :\ \alpha\mathbf{I} \preceq \mathbf{X} \preceq \beta\mathbf{I}\}} - \log\det X + \langle \Sigma + U, X \rangle.$$

in the variable $X \in \mathbf{S}_n$. Here, the objective function is already smooth on the feasible set, whenever $\alpha > 0$. Here too, the two projection steps can be computed by projecting the spectrum of the current iterate X, hence can be computed with complexity $O(n^3)$.

3. Numerical experiments

First-order methods for semidefinite optimization tradeoff a much lower cost per iteration with a much higher dependency on the target precision. This means that one cannot hope to obtain solutions up to a precision 10^{-8} that is routinely achieved by interior point algorithms. However, first-order methods do solve semidefinite optimization problems for which running even one iteration of interior point algorithms is numerically hopeless.

In Figure 1, using the data in [ABN+99], with $\rho = 1$, we plot CPU time to get a 10^2 decrease in duality gap. The computing times are summarized below. We notice that solving a dense semidefinite program of size 2000 up to a relative precision of 10^{-2} takes less than ten minutes on a quad-core computer with 2Gb of RAM.

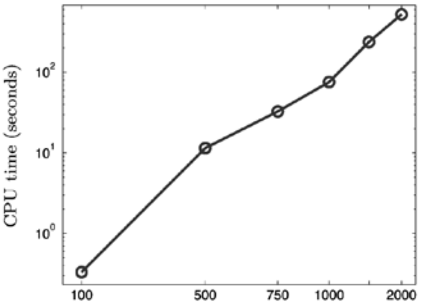


Fig. 1. CPU time for solving problem (1) on covariance matrices of increasing dimension n, formed using a gene expression data set.

n	CPU time (secs)
100	0 m 01 s
500	0 m 11 s
1000	1 m 16 s
2000	9 m 41 s

References

[ABN+99] A. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, *Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays*, Cell Biology **96** (1999), 6745–6750.

[Ali95] F. Alizadeh, *Interior point methods in semidefinite programming with applications to combinatorial optimization*, SIAM Journal on Optimization 5 (1995), 13–51.

[BEGd07] Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont, *Model selection through sparse maximum likelihood estimation*, ICML 2006.

[BV04] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge University Press, 2004.

[d'A05] A. d'Aspremont, *Smooth optimization with approximate gradient*, ArXiv: math.OC/0512344 (2005).

[dBEG06] A. d'Aspremont, O. Banerjee, and L. El Ghaoui, *First-order methods for sparse covariance selection*, SIAM Journal on Matrix Analysis and its Applications, 30(1), pp. 56-66, February 2008.

[dEGJL07] A. d'Aspremont, L. El Ghaoui, M.I. Jordan, and G. R. G. Lanckriet, *A direct formulation for sparse PCA using semidefinite programming*, SIAM Review 49 (2007), no. 3, 434–448.

[GW95] M.X. Goemans and D.P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. ACM 42 (1995), 1115–1145.

[HR00] C. Helmberg and F. Rendl, *A spectral bundle method for semidefinite programming*, SIAM Journal on Optimization 10 (2000), no. 3, 673–696.

[LCB+02] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, *Learning the kernel matrix with semi-definite programming*, 19th International Conference on Machine Learning (2002).

[Lew03] A.S. Lewis, *The mathematics of eigenvalue optimization*, Math. Program., Ser. B 97 (2003), 155–176.

[LNM07] Z. Lu, A. Nemirovski, and R.D.C. Monteiro, *Large-scale semidefinite programming via a saddle point Mirror-Prox algorithm*, Mathematical Programming 109 (2007), no. 2, 211–237.

[Mit03] H. D. Mittelmann, *An independent benchmarking of SDP and SOCP solvers*, Mathematical Programming Series B 95 (2003), no. 2, 407–430.

[MVL03] C. Moler and C. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Review 45 (2003), no. 1, 3–49.

[Nem04] A. Nemirovski, *Prox-method with rate of convergence O(1/T) for variational inequalities with lipschitz continuous monotone operators and smooth convexconcave saddle point problems*, SIAM Journal on Optimization 15 (2004), no. 1, 229–251.

[Nes05] Y. Nesterov, *Smooth minimization of non-smooth functions*, Mathematical Programming 103 (2005), no. 1, 127–152.

[Nes07] Y. Nesterov, *Smoothing technique and its applications in semidefinite optimization*, Mathematical Programming 110 (2007), no. 2, 245–259.

[NN94] Y. Nesterov and A. Nemirovskii, *Interior-point polynomial algorithms in convex programming*, Society for Industrial and Applied Mathematics, Philadelphia, 1994.

[Ous00] F. Oustry, *A second-order bundle method to minimize the maximum eigenvalue function*, Mathematical Programming 89 (2000), no. 1, 1–33.

[OW95] M. L. Overton and Robert S. Womersley, *Second derivatives for optimizing eigenvalues of symmetric matrices*, SIAM J. Matrix Anal. Appl. 16 (1995), no. 3, 697–718.

[SBXD05] J. Sun, S. Boyd, L. Xiao, and P. Diaconis, T*he fastest mixing Markov process on a graph and a connection to a maximum variance unfolding problem*, SIAM Review (2005).

[WS06] K.Q. Weinberger and L.K. Saul, *Unsupervised Learning of Image Manifolds by Semidefinite Programming*, International Journal of Computer Vision 70 (2006), no. 1, 77–90.

continued from page 8

Technical Editors

The review and testing of software is carried out by a board of Technical Editors, with guidance from the General Editor. The TE board has access to hardware/software platforms run by the journal to aid in the review process.

Software to review is assigned to a TE by one of the Area Editors or by the Editor-in-Chief. The review can be carried out by the TE, or a referee can be contacted. The identity of the assigned TE is not revealed to the author of the software. The TE may produce a public report describing the tests that were carried out; the report will be made available as supplementary material on the journal's web site. The initial TE board is made up of the following members.

- Tobias Achterberg, ILOG
- Erling D. Andersen, MOSEK ApS
- David Applegate, AT&T Research
- Oliver Bastert, Dash Optimization
- Pietro Belotti, Lehigh University
- Hande Y. Benson, Drexel University
- Andreas Bley, Konrad-Zuse-Zentrum Berlin
- Brian Borchers, New Mexico Tech
- Jordi Castro, Universitat Polit`ecnica de Catalunya
- Daniel Espinoza, University of Chile
- Armin Fügenschuh, TU Darmstadt
- Andreas Grothey, University of Edinburgh
- Zonghao Gu, Gurobi Optimization
- William Hart, Sandia National Laboratories
- Keld Helsgaun, Roskilde University
- Benjamin Hiller, Konrad-Zuse-Zentrum Berlin
- Leonardo B. Lopes, University of Arizona
- Todd S. Munson, Argonne National Laboratory
- Dominique Orban, Ecole Polytechnique de Montr´eal
- Ted Ralphs, Lehigh University
- Mohit Tawarmalani, Purdue University
- Stefan Vigerske, Humboldt-Universit¨at Berlin
- Richard A. Waltz, University of Southern California

6 Links for Further Information

- MPC Home Page mpc.zib.de
- MPS Journals www.mathprog.org/mps_pubs.htm
- Springer's MPC Page www.springer.com/math/journal/12532
- Notes on MPC www.isye.gatech.edu/˜wcook/mpc/
- MPC will use the Open Journal Systems (OJS) from Simon Fraser University pkp.sfu.ca/ojs

Nash equilibria computation via smoothing techniques

Javier Peña

October, 2008

1 Introduction

A *sequential game* is a mathematical model of the interaction of multiple self-interested players in dynamic stochastic environments with limited information. Poker is a widely popular example of these types of games. Unlike other popular sequential games such as chess or checkers, poker is a game of *imperfect information*. Consequently, speculation and counter-speculation are inherent features of the game and make the computation of optimal strategies a highly non-trivial task. Optimal strategies must necessarily include tactics such as bluffing and slow playing. Indeed, poker has been identified as an central challenge in artificial intelligence [2] and has become a topic of active research [1,6,7,9].

A fundamental solution concept for sequential games is *Nash equilibrium*, which is a simultaneous choice of strategies for all players so that each player's choice is optimal given the other players' choices. For a two-person, zero-sum sequential game, the Nash equilibrium problem has the following saddle-point formulation:

$$\min_{\mathbf{x}\in Q_1}\max_{\mathbf{y}\in Q_2}\langle \mathbf{y}, A\mathbf{x}\rangle = \max_{\mathbf{y}\in Q_2}\min_{\mathbf{x}\in Q_1}\langle \mathbf{y}, A\mathbf{x}\rangle. \qquad (1)$$

Here the sets Q_1, Q_2 are polytopes associated to the possible sequences of moves of the players, and A is Player 2's payoff matrix [3,8,14,15]. The saddle-point problem (1) can be cast as a linear program, but the resulting formulation is prohibitively large for most interesting games. For instance, the payoff matrix A in (1) for limit Texas Hold'em poker has dimension of order $10^{14}\times10^{14}$ and contains more than 10^{18} non-zero entries. Problems of this magnitude are far beyond the capabilities of state-of-the-art general-purpose linear programming solvers [4,5]. On the other hand, the equilibrium problem (1) possesses a great deal of structure that makes it particularly well-suited for Nesterov's smoothing techniques. The three main structural features are the saddle-point formulation, the combinatorial structure of the sets Q_1, Q_2, and a natural factorization of the payoff matrix A. As the sections below explain, these features are nicely compatible with Nesterov's smoothing technique. By taking advantage of these structural properties, we have computed near-equilibria for sequential games whose linear programming formulation would require about three hundred million variables and constraints, and over four trillion non-zeros entries. The computed near-equilibria have been instrumental in the design of competitive automatic poker players, including the winner of the 2008 AAAI annual poker competition.

2 Smoothing technique for saddle-point problems

The saddle-point problem (1) can be written as

$$\min_{\mathbf{x}\in Q_1} f(\mathbf{x}) = \max_{\mathbf{y}\in Q_2} \phi(\mathbf{y}),$$

where

$$f(\mathbf{x}) := \max_{\mathbf{y}\in Q_2}\langle \mathbf{y}, A\mathbf{x}\rangle \text{ and } \phi(\mathbf{y}) := \min_{\mathbf{x}\in Q_1}\langle \mathbf{y}, A\mathbf{x}\rangle.$$

The functions f and ϕ are non-smooth convex and concave respectively. As Nesterov points out in the previous article, the max-form of f and min-form of ϕ can be readily used to construct smooth approximations. More precisely, assume d_i is a non-negative and strongly convex function on Q_i for i = 1, 2. Such a function is called a *prox-function* for Q_i. For a given $\mu > 0$ the functions

$$f_\mu(\mathbf{x}) = \max_{\mathbf{y}\in Q_2}\{\langle \mathbf{y}, A\mathbf{x}\rangle - \mu d_2(\mathbf{y})\} \text{ and}$$
$$\phi_\mu(\mathbf{y}) := \min_{\mathbf{x}\in Q_1}\{\langle \mathbf{y}, A\mathbf{x}\rangle + \mu d_1(\mathbf{x})\} \qquad (2)$$

are smooth with Lipchitz gradients of order $O(1/\mu)$ and satisfy

$$0 \le f_\mu(\mathbf{x}) - f(\mathbf{x}) \le \mu D_2 \ \forall \mathbf{x}\in Q_1, \text{ and}$$
$$0 \le \phi_\mu(\mathbf{y}) - \phi(\mathbf{y}) \le \mu D_1 \ \forall \mathbf{y}\in Q_2,$$

where $D_i = \max\{d_i(\mathbf{u}) : \mathbf{u} \in Q_i\}$. By applying an optimal gradient method to the smooth approximations f_μ ,ϕ_μ , Nesterov [11,12] devised an algorithm that computes $\mathbf{x} \in Q_1, \mathbf{y} \in Q_2$ such that

$$0 \le f(\mathbf{x}) - \phi(\mathbf{y}) \le \varepsilon$$

in $O(1/\varepsilon)$ first-order iterations. Each iteration requires only a few elementary operations, some matrix-vector multiplications involving A, and the solution of some subproblems of the form

$$\max\{\langle \mathbf{s},\mathbf{u}\rangle - d_i(u) : \mathbf{u} \in Q_i\} \qquad (3)$$

for i = 1,2.

3 Nice prox-functions

In order for the above smoothing technique to be an implementable algorithm, problem (3) must be easily computable since it has to be solved several times at each iteration. We say that a prox-function d_i for Q_i is *nice* if the solution to problem (3) is easily computable, for example via a closed-form expression.

The polytopes Q_1, Q_2 arising in the Nash equilibrium problem (1) encode the *behavior strategies* of the players in the sequential game [13–15]. A behavior strategy for Player i prescribes a probability distribution over the choices available to Player i at every state of the game where it is Player i's turn to make a move. For games in strategic normal form, there is no sequential component and the sets Q_1, Q_2 are simplexes. In this case each element of Q_i is a probability distribution over the set of pure strategies available to Player i. For sequential games

The author would like to acknowledge support from NSF grant CCR-0830533, and a research grant from the Tepper School of Business.

J. Peña, Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA, USA
e-mail: jfp@andrew.cmu.edu

in extensive form, the sets Q_1,Q_2 are the sets of *realization plans* of the players. A realization plan is a concise encoding of a behavior strategy in terms of the possible *sequences of moves* of the player [14,15]. Sets of realization plans are polytopes that can be seen as a generalization of simplexes. They are obtained by recursive application a certain *branching* operation that encapsulates the relationship between consecutive sequences of moves [8,14,15].

A crucial ingredient in the implementation of a first-order smoothing algorithm for (1) is the construction of nice prox-functions for the sets of realization plans Q_1,Q_2. Hoda, Gilpin and Peña [8] provide a generic template that constructs nice prox-functions for Q_1,Q_2 using as building blocks any given family of nice prox-function for simplexes. In our numerical implementation, we have used the entropy function $\ln m+\sum_{i=1}^m x_i \ln x_i$ and the Euclidean distance function $½\sum_{i=1}^m(x_i -1/m)^2$ as building blocks. Both of these are families of nice prox-functions for simplexes [10,11]. In our computational experiments we obtained consistently faster convergence with the prox-functions induced by the entropy function.

4 Concise representation of the payoff matrix

The payoff matrix A in poker games has a diagonal block structure where each block in turn has a natural factorization. For instance, for a four-round poker game, the payoff matrix can be written as

$$A = \begin{bmatrix} F_1 \otimes B_1 & & & \\ & F_2 \otimes B_2 & & \\ & & F_3 \otimes B_3 & \\ & & & F_4 \otimes B_4 + S \otimes W \end{bmatrix}.$$

The matrices F_i correspond to sequences of moves in round i that end with a *fold*. The matrix S correspond to sequences of moves that end with a *showdown*. The matrices $\boldsymbol{B}_i$ encode the betting structure in round i. Finally, the matrix W encode the win/lose/draw information determined by poker hand ranks.

We take advantage of this factorization to avoid forming the matrix A explicitly. Instead, we construct subroutines that compute the matrix-vector products $\mathbf{x} \mapsto \mathbf{Ax}$ and $\mathbf{y} \mapsto \mathbf{A^Ty}$ as needed in the smoothing algorithm. This concise representation of the payoff matrix yields dramatic savings in terms of the amount of space needed to store a problem instance. In particular, the concise representation for the largest game that we have handled so far requires about 40GB of memory [3]. The additional memory overhead required by the smoothing algorithm is essentially negligible. By contrast, an explicit sparse representation of this problem would require over 80,000GB of memory. A general-purpose linear programming solver, such as an interior-point algorithm, would require a substantial additional amount of memory throughout its execution.

In addition to the savings in memory requirements, the above concise representation of the payoff matrix A can be used for parallel computation. A parallel implementation of the matrix-vector subroutines achieves nearly a linear speedup [3]. This in turn has an immediate impact on the overall performance of the smoothing algorithm because the matrix-vector products are the main bottleneck at each iteration.

References

1. Bard, N., Bowling, M.: Particle filtering for dynamic agent modelling in simplified poker. In: Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI), pp. 515–521 (2007)
2. Billings, D., Peña, L., Schaeffer, J., Szafron, D.: The challenge of poker. Artificial Intelligence 134(1-2), 201–240 (2002). Special Issue on Games, Computers and Artificial Intelligence
3. Gilpin, A., Hoda, S., Peña, J., Sandholm, T.: Gradient-based algorithms for finding Nash equilibria in extensive form games. In: Workshop on Internet and Network Economics (WINE-07) (2007)
4. Gilpin, A., Sandholm, T.: Optimal Rhode Island Hold'em poker. In: M. Veloso, S. Kambhampati (eds.) 20st National Conference on Artificial Intelligence (AAAI-05), pp. 1684–1685 (2005)
5. Gilpin, A., Sandholm, T.: Finding equilibria in large sequential games of imperfect information. In: Proceedings, ACM Conference on Electronic Commerce (EC'06), Ann Arbor, MI (2006)
6. Gilpin, A., Sandholm, T.: Expectation-based versus potential-aware automated abstraction in imperfect information games: An experimental comparison using poker. In: D. Fox, C.P. Gomes (eds.) Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008, pp. 1454–1457. AAAI Press (2008)
7. Gilpin, A., Sandholm, T., Sørensen, T.B.: GS3 and Tartanian: game theory-based headsup limit and no-limit texas hold'em poker-playing programs. In: AAMAS (Demos), pp. 1647–1648. IFAAMAS (2008)
8. Hoda, S., Gilpin, A., Peña, J.: Smoothing techniques for computing Nash equilibria of sequential games. Tech. rep., Carnegie Mellon University (2008)
9. Miltersen, P.B., Sørensen, T.B.: A near-optimal strategy for a heads-up no-limit texas hold'em poker tournament. In: E.H. Durfee, M. Yokoo, M.N. Huhns, O. Shehory (eds.) 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007), Honolulu, Hawaii, USA, May 14-18, 2007, p. 191. IFAAMAS (2007)
10. Nemirovski, A.: Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. SIAM Journal on Optimization 15(1), 229–251 (2004)
11. Nesterov, Y.: Excessive gap technique in nonsmooth convex minimization. SIAM J. on Optim. 16(1), 235–249 (2005)
12. Nesterov, Y.: Smooth minimization of non-smooth functions. Math Program. 103(1), 127–152 (2005)
13. Osborne, M., Rubinstein, A.: A Course in Game Theory. MIT Press, Cambridge, MA (1994)
14. Stengel, B.V.: Efficient computation of behavior strategies. Games and Economic Behavior 14, 220–246 (1996)
15. Stengel, B.V.: Equilibrium computation for games in strategic and extensive form. In: N. Nisan, T. Roughgarden, E. Tardos, V.V. Vazirani (eds.) Algorithmic Game Theory. Cambridge University Press (2007)

D.R. FULKERSON PRIZE

Call for nominations

The Fulkerson Prize Committee invites nominations for the Delbert Ray Fulkerson Prize, sponsored jointly by the Mathematical Programming Society and the American Mathematical Society. Up to three awards are presented at each (triennal) International Symposium of the MPS. The Fulkerson Prize is for outstanding papers in the area of discrete mathematics. The Prize will be awarded at the 20th International Symposium on Mathematical Programming to be held in Chicago, August 23-29, 2009.

Eligible papers should represent the final publication of the main result(s) and should have been published in a recognized journal, or in a comparable, well-refereed volume intended to publish final publications only, during the six calendar years preceding the year of the Symposium (thus, from January 2003 through December 2008). The prizes will be given for single papers, not series of papers or books, and in the event of joint authorship the prize will be divided.

The term 'discrete mathematics' is interpreted broadly and is intended to include graph theory, networks, mathematical programming, applied combinatorics, applications of discrete mathematics to computer science, and related subjects. While research work in these areas is usually not far removed from practical applications, the judging of papers will be based only on their mathematical quality and significance.

Further information about the Fulkerson Prize including previous winners can be found at www.mathprog.org/prz/fulkerson.htm and at www.ams.org/prizes/fulkerson-prize.html.

The Fulkerson Prize Committee consists of Bill Cook, Georgia Tech, chair, Michel Goemans, MIT and Danny Kleitman, MIT.

Please send your nominations (including reference to the nominated article and an evaluation of the work) by January 15th, 2009 to the chair of the committee. Electronic submissions to bico@isye.gatech.edu are preferred.

William Cook
Industrial and Systems Engineering
Georgia Tech
765 Ferst Drive
Atlanta, Georgia 30332-0205
USA

e-mail: bico@isye.gatech.edu

Beale-Orchard-Hays Prize 2009—Call for nominations

Nominations are invited for the 2009 Beale-Orchard-Hays Prize for excellence in computational mathematical programming.

The Prize is sponsored by the Mathematical Programming Society, in memory of Martin Beale and William Orchard-Hays, pioneers in computational mathematical programming. Nominated works must have been published between Jan 1, 2006 and Dec 31, 2008, and demonstrate excellence in any aspect of computational mathematical programming. Computational mathematical programming includes the development of high-quality mathematical programming algorithms and software, the experimental evaluation of mathematical programming algorithms, and the development of new methods for the empirical testing of mathematical programming techniques. Full details of prize rules and eligibility requirements can be found at www.mathprog.org/prz/boh.htm

The 2009 Prize will be awarded at the awards session of the International Symposium on Mathematical Programming, to be held August 23-28, 2009, in Chicago, Illinois, USA. Information about the Symposium can be found at www.ismp2009.org
The 2009 Prize Committee consists of

Erling Andersen, *Mosek*
Philip Gill, *University of California San Diego*
Jeff Linderoth, *University of Wisconsin Madison*
Nick Sahinidis (chair), *Carnegie Mellon University*

Nominations can be submitted electronically or in writing, and should include detailed publication details of the nominated work. Electronic submissions should include an attachment with the final published version of the nominated work. If done in writing, submissions should include four copies of the nominated work. Supporting justification and any supplementary material are strongly encouraged but not mandatory. The Prize Committee reserves the right to request further supporting material and justification from the nominees.

Nominations should be submitted to:

Nick Sahinidis
Department of Chemical Engineering
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
USA
e-mail: sahinidis[at]cmu.edu

The deadline for receipt of nominations is January 15, 2009.

2009 ISMP

20th International Symposium on Mathematical Programming Chicago, August 23-29, 2009

The 20th International Symposium on Mathematical Programming will take place August 23-29, 2009 in Chicago, Illinois. The meeting will be held at the University of Chicago's Gleacher Center and the Marriott Downtown Chicago Magnificent Mile Hotel. Festivities planned for the conference include the opening session in Chicago's Orchestra Hall, home of the Chicago Symphony Orchestra, the conference banquet at the Field Museum, Chicago's landmark natural history museum, and a celebration of the 60th anniversary of the Zeroth ISMP Symposium.

The invited speakers are
Eddie Anderson, *University of Sydney*
Friedrich Eisenbrand, *EPFL*
Matteo Fischetti, *University of Padova*
Pablo Parrilo, *MIT*
Martin Skutella, T*echnische Universität Berlin*
Éva Tardos, *Cornell University*
Shuzhong Zhang, *Chinese University of Hong Kong*
Mihai Anitescu, *Argonne National Lab*
András Frank, *Eötvös Loránd University*
Jong-Shi Pang, *UIUC*
Andrzej Ruszczynski, *Rutgers University*
David Shmoys, *Cornell University*
Paul Tseng, *University of Washington*

Papers on all theoretical, computational and practical aspects of mathematical programming are welcome. The program clusters and their organizers have been announced. Parties interested in organizing a session are encouraged to contact the cluster chairs.

Combinatorial Optimization	András Frank, Tom McCormick
Integer and Mixed-Integer Programming	Andrea Lodi, Robert Weismantel
Nonlinear Programming	Philip Gill, Philippe Toint
Nonlinear Mixed-Integer Progamming	Sven Leyffer, Andreas Wächter
Complementarity and Variational Inequalities	Masao Fukushima, Danny Ralph
Conic Programming	Kim-Chuan Toh
Nonsmooth and Convex Optimization	Michael Overton, Marc Teboulle
Stochastic Optimization	Shabbir Ahmed, David Morton
Robust Optimization	Aharon Ben-Tal
Global Optimization	Christodoulos A. Floudas, Nick Sahinidis
Logistics and Transportation	Xin Chen, Georgia Perakis
Game Theory	Asu Ozdaglar, Tim Roughgarden.
Telecommunications and Networks	Martin Skutella
Approximation Algorithms	Cliff Stein, Chandra Chekuri
Optimization in Energy Systems	Andy Philpott, Claudia Ságastizabal
PDE-Constrained Optimization	Matthias Heinkenschloss, Michael Hintermüller
Derivative-Free and Simulation-Based Optimization	Jorge Moré, Katya Scheinberg
Sparse Optimization	Michael Saunders, Yin Zhang
Finance and Economics	Tom Coleman, Kenneth Judd
Implementations and Software	Erling Andersen, Michal Kočvara
Variational Analysis	Boris Mordukhovich, Shawn Wang

Registration and hotel information will be available by the end of November. Further information about the symposium can be found on the conference Web site, www.ismp2009.org.

O P T I M A

MATHEMATICAL PROGRAMMING SOCIETY

UF UNIVERSITY of FLORIDA

Center for Applied Optimization
401 Weil Hall
P.O. Box 116595
Gainesville, FL 32611-6595 USA

FIRST CLASS MAIL

PUBLISHED BY THE
MATHEMATICAL PROGRAMMING SOCIETY &
University of Florida

Journal contents are subject to change by the publisher.