8 Some Problems in Nonlinear Programming

Ya-xiang Yuan

State Key Laboratory of Scientific and Engineering Computing
Institute of Computational Mathematics and Scientific/Engineering Computing
Chinese Academy of Sciences, Beijing 100080, China

Abstract: In this paper we give an review on convergence problems of unconstrained optimization algorithms, including line search algorithms and trust region algorithms. Recent results on convergence of conjugate gradient methods are discussed. Some well-known convergence problems of variable metric methods and recent efforts made on these problems are also presented.

Key Words: unconstrained optimization, constrained optimization, convergence, line search, trust region.

1 INTRODUCTION

In this paper, we give some problems about nonlinear programming. These problems are interesting problems.

Unconstrained optimization is to minimize a nonlinear function $f(x)$, which can be written as

$$\min_{x \in \Re^n} f(x). \tag{1.1}$$

2 THE PROBLEMS

Problem 1. *Is there a better steepest descent method?*

The steepest descent method is the simpliest method among all optimization methods that use gradients. At each step, the negative gradient direction is used as the search direction, namely

$$x_{k+1} = x_k + \alpha_k(-g_k), \tag{2.1}$$

where $\alpha_k > 0$ is the stepsize and $g_k = g(x_k) = \nabla f(x_k)$. The advantages of the steepest descent method are less storage requirement, easy to implement and robust convergence even for ill-conditioned problems. However, a serve disadvantage of the steepest descent method is that it converges very slowly when exact line searches are used[11]. Indeed, zig-zags normally will happen for steepest descent method with exact line searches. It A very interesting and surprising result was discovered by Barzilai and Borwei[1] about the steepest descent method. They suggested that the steplength of the exact line search in the current iteration, be used in the next iteration. By this simple modification, they proved that method will converge R-superlinearly for two-dimensional strictly convex quadratic functions. However, for higher dimension whether the BB method is superlinearly convergent is unknown. Another related question is whether we can use the function and gradient values at the previous two iterations to define the stepsize.

Problem 2. *How can we use function values to construct new conjugate gradient methods?*

Conjugate gradient methods use a linear combination of the steepest descent direction and the previous search direction as the search direction in the current iteration. A detailed discussion on conjugate gradient methods can be found in [4] and [2]. All conjugate gradient methods set the search direction by

$$d_{k+1} = -g_{k+1} + \beta_k d_k. \tag{2.2}$$

Different conjugate gradient methods use different formular for β_k. Detailed discussions on conjugate gradient methods can be found in [4] and [3]. The definition of β_k depends, generally, on the following values: $||g_k||$, $||g_{k+1}||$, $g_k^T g_{k+1}$, $g_k^T d_k$ and $g_{k+1}^T d_k$. These values are not directly related to the function reduction in the previous iteration: $f(x_k) - f(x_{k+1})$. Therefore it would interesting to ask whether it is possible to obtain a formular for β_k which depends on $f_k - f_{k+1}$. For example, a possibility could be

$$\beta_k = \frac{g_{k+1}^T y_k}{(f_k - f_{k+1})/\alpha_k - d_k^T g_k/2}. \tag{2.3}$$

Problem 3. *Three dimensional Minimization Model*

A two dimensional minimization model was proposed by Yuan and Stoer, which defines the next iterate point x_{k+1} by the minimizer of the quadratic model in the 2-dimensional subspace:

$$S_k = x_k + SPAN\{-g_k, d_{k-1}\}. \tag{2.4}$$

A natural extension of this idea is to consider the 3-dimensional subspace:

$$S_k = x_k + SPAN\{-g_k, d_{k-1}, y_{k-1}\}. \tag{2.5}$$

The present of y_{k-1} is not a surprise as for the one-step limited memory BFGS method, the search direction d_{k+1} can be expressed as the linear combination of $-g_k$, d_{k-1} and y_{k-1}. Similar to the 2-dimensional case, the main task is to estimate the values

$$g_k^T \nabla^2 f(x_k) g_k, \qquad y_{k-1}^T \nabla^2 f(x_k) y_{k-1}. \tag{2.6}$$

The technique used in Yuan and Stoer can still be used, which is to replace the square of a cosine value by its meanvalue 1/2. A possible way to estimate $g_k^T \nabla^2 f(x_k) g_k$ is to study the projection of g_k to the previous directions such as d_{k-1} and d_{k-2}.

Problem 4. *Convergence of DFP method for convex functions with inexact line search.*

This is the long time standing problem in the convergence theory of nonlinear programming. The DFP method is the very first quasi-Newton method, though it is not the best one. It is widely regarded that the BFGS method may be the best quasi-Newton method, even we do not have convincing theoretical proofs. There are some studies about the differences between the BFGS method and the DFP method, for example see Powell[15].

For exact line searches, Dixon[9] proved that all quasi-Newton methods in Huang's family will generate identical points, therefore the convergence problem for quasi-Newton methods is solved as Powell[13] proved the convergence of DFP method.

If line searchers are not exact, Powell[14] proved the convergence of the BFGS method assuming the line search conditions are

$$f(x_k + \alpha_k d_k) \le f(x_k) + c_1 \alpha_k d_k^T g_k, \tag{2.7}$$

$$d_k^T \nabla f(x_k + \alpha_k d_k) \geq c_2 d_k^T \nabla f(x_k), \tag{2.8}$$

where $c_1 \leq c_2$ are two positive constants in $(0, 1)$. The question is as follows: whether the DFP method convergence if the line search conditions are the above two inequalities? Yuan[21] showed that if the DFP method will converge if the normals of the gradients $g_k||$ are monotone with respect to k. But whether the method ensures convergence without this artificial assumption is still unknown.

Problem 5. *What is the exact convegence rate of quasi-Newton methods?*

Assume that the objective function is uniformly convex, and exact line searches are used at every iteration. Quasi-Newton methods in the Broyden's family will generate identical points. And from the famous Dennis-Moré theory [8], we know that the iterates will converge to the solution x^* Q-superlinearly in the sense that

$$\lim_{k\to\infty} \frac{||x_{k+1} - x^*||}{||x_k - x^*||} = 0. \tag{2.9}$$

However, if we consider all twice continously differential functions, for any given positive number ϵ, an example can be constructed to show that

$$||x_{k+1} - x^*|| = O(||x_k - x^*||^{1+\epsilon}) \tag{2.10}$$

may fail. Therefore the least Q-order of convergence is only 1. It is interesting to know the exact convergence rate, namely how quick the ratio

$$\frac{||x_{k+1} - x^*||}{||x_k - x^*||} \tag{2.11}$$

converges to zero. Assume that the objective function is three times continuously differentiable, the best result is obtained by Powell[16]

$$||x_{k+1} - x^*|| = O(||x_k - x^*|| ||x_{k-n+1} - x^*||). \tag{2.12}$$

Under the condition that

$$(s_k/||s_k||, s_{k-1}/||s_{k-1}||, ..., s_{k-n+1}/||s_{k-n+1}||)^{-1} \tag{2.13}$$

is uniformly bounded, Ritter showed that

$$||x_{k+1} - x^*|| = O(||x_k - x^*|| ||x_{k-1} - x^*||). \tag{2.14}$$

When $n = 2$, (2.12) and (2.14) are the same, which is the best result that we can expect because it is possible that

$$||x_{k+1} - x^*|| = ||x_k - x^*|| ||x_{k-1} - x^*|| \tag{2.15}$$

for all k. For $n > 2$, (2.14) is stronger than (2.12). The question is whether we could prove (2.14) without the uniformly boundedness assumption on (2.13)?

Problem 6. *How can we construct a direct method based on an explicit formular using f_i and $s_i (i = k, k-1, .., k-m)$?*

Direct methods are those use only function values. A widely used direct method is the conjugate direction method[11]. Recently Powell proposed a trust region type quadratic approximation method, which computes the trial step by minimizing a quadratic model within a trust region ball. The quadratic model is an approximation to the objective function using interpolation conditions in the previous iterations. i The general form for the trial step or the search direction (depending whether we use trust region or line search) would be

$$d(x_k, f_k, f_{k-1}, ..., f_{k-m}, s_{k-1}, ..., s_{k-m}) \tag{2.16}$$

For a line search type method, we would require that

$$\min \|d(x_k, f_k, f_{k-1}, ..., f_{k-m}, s_{k-1}, ..., s_{k-m}) - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)\|_W, \tag{2.17}$$

where $\|.\|_W$ is some norm in $\Re^n$. For a trust region method, we would require

$$\min \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d \tag{2.18}$$

$$s.t. \quad d = d(x_k, f_k, f_{k-1}, ..., f_{k-m}, s_{k-1}, ..., s_{k-m}), \quad \|d\| \le \Delta_k. \tag{2.19}$$

The unusual issue here is that the values $\nabla f(x_k)$ and $\nabla^2 f(x_k)$ are not available. For the first subproblem, a possible way is to choose the search direction in the form

$$d_k = \sum_{i=1}^{m} \beta_i^{(k)} s_{k-i}. \tag{2.20}$$

The coefficients $\beta_i^{(k)}$ should be functions of s_{k-1}, ..., s_{k-m}, f_k, f_{k-1}, ..., f_{k-m}. The question is what are the good formulae for $\beta_i^{(k)}$? This approach requires $m \ge n$, because if $m < n$, all search directions will be in the subspace spaned by the first initial m directions. Therefore, for $m < n$, a reasonable way to define the search direction should be

$$d_k = \hat{d}_k + \sum_{i=1}^{m} \beta_i^{(k)} s_{k-i}, \tag{2.21}$$

where $\hat{d}_k$ is a random vector.

Problem 7. *What is the best way to use null space technique for solving nonlinear equations and nonlinear least squares problems?*

Nonlinear systems of equations and nonlinear least squares problems are very common in practice. For our discussions here, we just consider the case of nonlinear systems of equations. Assume that we want to solve

$$F(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_m(x) \end{pmatrix} = 0. \tag{2.22}$$

For linear problems, it is easy to see that we should take the Newton's step, namely we can let

$$F(x_k) + J_k^T d_k = 0 \tag{2.23}$$

where J_k is the Jacobian matrix. However, for nonlinear problems, either the Levenberg-Marquardt method or a trust region approach will set a shorter trial step. Now we suppose that in our problem, some $F_i(x)$ are linear functions and the others are nonlinear. We can also view our problem in this way if some equations are highly nonlinear and the rest are less nonlinear. For simplicity, we assume that $F_i(x)(i = 1, .., I)$ are linear functions and $F_i(x)(i = I+1, ..., m)$ are nonlinear function. In this case, in a trust region method, we would like to d_k be defined by

$$\min \sum_{i=I+1}^{m} (F_i(x_k) + d^T \nabla F_i(x_k))^2 \tag{2.24}$$

subject to

$$F_i(x_k) + d^T \nabla F_i(x_k) = 0 \quad (i = 1, ..., I). \tag{2.25}$$

This is in fact to search on the null space of a subset of equations. In practice, how to automatically identify the subset of the equations that are more "linear-like" is the key issue in this approach. Instead using the nonlinearality to define the set I, we can also use the residuals to selecting equations to be treated as constraints. Suppose that all equations are nonlinear, it might be reasonable to devide the equations into two group, those with small residuals and those with large residuals. Let I_k be the subset with small residuals, one may consider to obtain the search direction or trial step by solving the problem

$$\min \sum_{i \notin I_k} (F_i(x_k) + d^T \nabla F_i(x_k))^2 \tag{2.26}$$

subject to

$$F_i(x_k) + d^T \nabla f_i(x_k) = 0 \qquad (i \in I_k). \tag{2.27}$$

Problem 8. *Null Space Technique*

Let d_k be a trial step (for a trust region algorithm) or a search direction (for a line search direction) at the $k-th$ iteration in a numerical method for constrained optimization We can always decomposite the vector d_k into two parts, one in the null space and the other in the range space. Namely, we can write

$$d_k = h_k + v_k \tag{2.28}$$

where h_k is the null space vector, which is also called the horizontal step, v_k is the range space vector, which is also called the vertical step. The Marotos Effect may reject the step d_k even when it is a superlinearly convergent step. One remedy for the Marotos Effect is to take a second order correction step $\hat{d}_k$. For example, in equality constrained optimization, the second order step can be the minimum norm solution of the linearized constraints:

$$c(x_k + d_k) + J_k d = 0, \tag{2.29}$$

where J_k is the Jacobian matrix of $c(x)$ at x_k. It is easy to see that $\hat{d}_k$ is also a vertical step, which can also be denoted by $\hat{v}_k$. Therefore some algorithms would take two vertical steps and one horizontal step in one single iteration:

$$x_{k+1} = x_k + h_k + v_k + \hat{v}_k. \tag{2.30}$$

However, the vertical step is a Newton's step, thus it is quadratic convergent. But in the null space, we need to minimize the Lagrange function. Because normally we could not afford to compute the Hessian matrix of the Lagrange function, the horizontal step can not be a Newton step, which converges slower than quadratic. Therefore, intuitively every iteration should take two steps in the null space and one step in the range step, namely

$$x_{k+1} = x_k + h_k + v_k + \hat{h}_k. \tag{2.31}$$

where $\hat{h}_k$ is a correction step in the null space. Indeed, Yuan[22] analyze the quadratic convergence of such a method. However, in the method studied in Yuan[22], the Jacobians of the constraints at the non-iterate point $x_k + h_k + v_k$ need to be computed. How to balance the computations in the null space and those in the range space and to maintain a fast convergence speed is an interesting problem to study.

Problem 9. *Celis-Dennis-Tapia Problem*

The Celis-Dennis-Tapia Problem is a subproblem arised in trust region algorithms for equality contrained optimization. The CDT problem has the following form:

$$\min_{d \in \Re^n} g^T d + \frac{1}{2} d^T B d \tag{2.32}$$

subject to

$$\|A^T d + c\|_2 \leq \xi \tag{2.33}$$

$$\|d\|_2 \leq \Delta, \tag{2.34}$$

where $g \in \Re^n$, $B \in \Re^{n \times n}$ symmetric, $A \in \Re^{n \times m}$, $c \in \Re^m$, $\xi \geq 0$ and $\Delta > 0$.

Let d^* be a solution of the CDT problem (2.32)-(2.34), it can be shown that there exist two multipliers λ and μ such that

$$(B + \lambda I + \mu A A^T) d^* = -(g + \mu A c), \tag{2.35}$$

and

$$\lambda(\Delta - \|d^*\|_2) = 0, \tag{2.36}$$

$$\mu(\xi - \|A^T d^* + c\|_2) = 0. \tag{2.37}$$

$$\|d^*\|_2 \leq \Delta, \qquad \|A^T d + c\|_2 \leq \xi. \tag{2.38}$$

Furthermore, the Hessian of the Lagrange function

$$H(\lambda, \mu) = B + \lambda I + \mu A A^T \tag{2.39}$$

has at most one negative eigenvalue.

For convex problems (when B is positive definite), we can solve the dual problem, which can be written as

$$\max -\frac{\lambda}{2}\Delta^2 + \frac{\mu}{2}(\|c\|^2 - \xi^2) - \frac{1}{2}(g + \mu A c)^T H(\lambda, \mu)^{-1}(g + \mu A c) \tag{2.40}$$

$$s.t. \qquad \lambda \geq 0, \qquad \mu \geq 0. \tag{2.41}$$

If the original problem is convex, i.e., B is positive definite, the above dual problem is to maximize a concave function in one quarter of the plane. In this case, the dual problem can be easily solved by truncated Newton's method proposed by Yuan[20] or the variable elimination method of Zhang[23]. For general nonconvex problems, Chen and Yuan[7] studies the location of the dual solution.

Problem 10. *The underline properties of the linear systems corresponding subproblems*

For a SQP method, the search virection d_k is a solution of the quadratic problem

$$\min g_k^T d + \frac{1}{2} d^T B_k d \tag{2.42}$$

subject to

$$c_k + A_k^T d = 0. \tag{2.43}$$

If B_k is the Hessian matrix of the Lagrange function, the SQP step is just Lagrange-Newton step, as it is the Newton's step for finding the stationary point of the Lagrange function[11]. Therefore, if we denote the multipliers of the QP problem (2.42)-(2.43) by η, the equality constrained QP is equivalent to the following linear system:

$$\begin{bmatrix} B_k & A_k \\ A_k^T & 0 \end{bmatrix} \begin{pmatrix} d \\ \eta \end{pmatrix} = - \begin{pmatrix} g_k \\ c_k \end{pmatrix} \tag{2.44}$$

Now consider the CDT subproblem. The solution d_k of the CDT subproblem would satisfy (2.35). Define $\eta = \mu(A^T d + c)$, we can see that

$$\begin{bmatrix} B + \lambda I & A \\ A^T & -\frac{1}{\mu} \end{bmatrix} \begin{pmatrix} d \\ \eta_k \end{pmatrix} = - \begin{pmatrix} g_k \\ c_k \end{pmatrix} \tag{2.45}$$

The Courant penlaty function

$$P(x, \sigma) = f(x) + \frac{1}{2}\sigma \|c(x)\|_2^2 \tag{2.46}$$

Consider the Newton's step for the above penalty function,

$$[\nabla^2 f(x_k) + \sigma A_k A_k^T + \sigma \sum_{i=1}^{m} c_i(x_k) \nabla^2 c_i(x_k)] d + \nabla f(x_k) + \sigma A_k c_k = 0 \tag{2.47}$$

Define $W_k = \nabla^2 f(x_k) + \sigma \sum_{i=1}^{m} c_i(x_k) \nabla^2 c_i(x_k)$, and $\sigma(A_k^T d + c_k) = \eta_k$, the above relation can be rewritten as

$$\begin{bmatrix} W_k & A_k \\ A_k^T & -\frac{1}{\sigma} \end{bmatrix} \begin{pmatrix} d \\ \eta_k \end{pmatrix} = - \begin{pmatrix} g_k \\ c_k \end{pmatrix} \tag{2.48}$$

We can see that the systems (2.44), (2.45) and (2.48) are similar systems. It is interesting to study the properties of these linear systems, and try to construct new methods by proposing different linear systems which have similar structures.

Problem 11. *Trust Region Algorithms that allows the trust region bound Converging to zero?*

It is usually that a trust region algorithm will not reduce its trust region bound if the ratio of the predicted reduction and the actual reduction is greater than a certain positive number. Therefore, under second order sufficient conditions, when the iterates are close the the solution, the trust region bound will never be reduced, which ensures that eventually the trust region constraint will be inactive. Consequently we can establish the local superlinear convergence result. This seems to be perfect and it looks like that there is no room to raise a question. However, two points motivate us to rethink this matter. First, the bounded away from zero of the trust region bound is a sufficient condition for the trust region constraint being inactive, but not a necessary condition. In fact, for any superlinearly convergent trial step d_k, as long as the trust region bound Δ_k satisfies

$$\Delta_k > \|d_k\|, \tag{2.49}$$

the trust region constraint $\|d\| \leq \Delta_k$ will be inactive. Thus, from (2.49) we can see that we can allow Δ_k converging to zero while maintaining the inactiveness of the trust region constraint. Secondly, for some singular problems it would be better to have a small trust region bound to avoid the iterate points wandering near the solution set. In fact, for singular nonlinear least square problems, Yamashita and Fukushima[19] suggest using the Levenberge-Marquardt method with the parameter chosen by $\|F(x_k)\|^2$. If we view the Levenberg-marquardt method in the framework of a trust region algorithm, a nonzero Levenberg-Marquardt parameter implies that the trust region constraint is active. Therefore for sigular problems, the activeness of the trust region constraint may not be a bad property. Actually, as long as the trust region bound Δ_k is greater than the distance from the current iterate x_k to the solution set X^*, the trust region constraint will not harm the local superlinear convergence. Therefore, it would be desirable to use the trust region bound as follows

$$\Delta_k = \mu_k \phi(x_k) \tag{2.50}$$

where $\phi(x_k)$ is some kind of approximation to the function $dist(x_k, X^*)$. The coefficient μ_k is updated from iteration to iteration. For unconstrained optimization problems, one possible choice is $\phi(x_k) = \|\nabla f(x_k)\|$. However, Powell [17] mentioned that this technique may slow the convergence for the case when $f(x) = x^4$. This simple example indicates that special attentions should be given to obtain a good approximation to the distance function $dist(x_k, X^*)$.

Hei[12] has studied the case where the trust region bounds are defined by

$$\Delta_{k+1} = R(\rho_k)\|d_k\| \tag{2.51}$$

where $R(.)$ is a certain monotonic function, ρ_k is the ratio between the actual reduction and the predicted reduction, and d_k is the trial step in the $k-$th iteration.

3 ACKNOWLEDGMENTS

This work is partially supported by National Natural Science Foundation of China grants 19525101, 19731010 and State Key Project 96-221-04-02-02.

References

[1] J. Barzilai and J.M. Borwein, "Two-point step size gradient methods", *IMA J. Numer. Anal.* 8(1988) 141-148.

[2] Y.H. Dai, *Analyses of Conjugate Gradient Methods* (in Chinese), Ph.D. Thesis, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, Beijing, 1997.

[3] Y.H. Dai, "Unified convergence analyses of nonlinear conjugate gradient methods", in: Y. Yuan, ed., *Numerical Linear Algebra and Optimization* (Science Press, Beijing, 2001), pp. 35-49.

[4] Y.H. Dai and Y. Yuan, *Nonlinear Conjugate Gradient Methods* (in Chinese), (Shanghai Sci and Tech Press, Shanghai, 2000).

[5] Y. Dai, J. Han, G. Liu, D. Sun, H. Yin and Y. Yuan, Convergence properties of nonlinear conjugate gradient methods, Research Report. ICMSEC-98-24, Institute of Comp. Math. and Scientific/Engineering Computing, Chinese Academy of Sciences, Beijing, China, 1998.

[6] Y. H. Dai and Y. Yuan, Convergence properties of the Fletcher-Reeves method, IMA J. Numer. Anal. 16 (1996):155–164.

[7] X.D. Chen and Y. Yuan, "On the local solutions of the CDT subproblems", *SIAM J. Optimization* 10(1999) 359-383.

[8] J.E. Dennis and J.J. Moré, "Quasi-Newton method, motivation and theory", *SIAM Review* 19(1977) 46-89.

[9] L.C.W. Dixon, "Variable metric algorithms: necessary and sufficient conditions for identical behavior of nonquadratical functions", *J. Opt. Theory Appl.* 10(1972) 34-40.

[10] R. Fletcher, A model algorithm for composite NDO problem, Math. Prog. Study, 17(1982):67-76. (1982a)

[11] R. Fletcher, Practical Methods of Optimization, 2nd edition, John Wiley and Sons, Chichester, 1987.

[12] L. Hei, *On the Extension of Trust Region Algorithms for Unconstrained Optimization* M.Sc. Thesis, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, Beijing, 2001.

[13] M.J.D. Powell, "On the convergence of the variable metric algorithm", *J. Inst. Maths. Appl.* 7(1971) 21-36.

[14] M.J.D. Powell, Some global convergence properties of a variable metric algorithm for minimization without exact line searches, in: R.W. Cottle and C.E. Lemke, eds., Nonlinear Programming, SIAM-AMS Proceeding vol. IX, SIAM publications, Philadelphia, 1976, pp. 53-72.

[15] M.J.D. Powell, "How bad ar the BFGS and DFP methods when the objective function is quadratic?", *Math. Prog.* 34(1986) 34-47.

[16] M.J.D. Powell, "On the rate of convergence of variable metric algorithms for unconstrained optimization", in: Z. Giesielki and G. Olech, eds., *Proceedings of the International Congress of Mathematicians* (Elsevier, New York, 1984) pp. 1525-1539.

[17] M.J.D. Powell, private communication, 2002.

[18] P. Wolfe, Convergence conditions for ascent methods, SIAM Rev. 11 (1969): 226–235.

[19] N. Yamashita and M. Fukushima, "On the rate of convergence of the Levenberg-Marquardt method", *Computing* (Supplement 15) (2001), 237-249.

[20] Y. Yuan, "A dual algorithm for minimizing a quadratic function with two quadratic constraints", *J. Computational Mathematics*, 9(1991) 348-359.

[21] Y. Yuan, Convergence of DFP algorithm, Science in China(Series A) 38(1995): 1281–1294.

[22] Y. Yuan, "A null space algorithm for constrained optimization", in: Z.C. Shi et.al. eds., *Advances in Scientific Computing* (Science Press, Beijing/New York, 2001) pp. 210-218.

[23] Y. Zhang, "Computing a Celis-Dennis-Tapia trust region step for equality constrained optimization", *Math. Prog.* 55(1992) 109-124.