Mathematica Balkanica

Mathematical Society of South-Eastern Europe
A quarterly published by
the Bulgarian Academy of Sciences – National Committee for Mathematics

---

Mathematica Balkanica

New Series Vol. 2, 1988, Fasc. 2-3

On Computationally Implementable Variants of the Method of Centers of Gravity

T. I. Encheva, A. Iu. Levin, D. L. Vandev

Presented by P. Kenderov

A class of convex programming algorithms, which are closely related to the method of centers of gravity, is discussed. Particular attention is paid to practically realizable methods, which do not enclose domains, where the extremal point is localized into simply structured domains. The considered versions are of importance when computation of the function and its gradient values is very laborious. The problem of minimizing a quasi-convex function is separately considered. A data structure for an effective maintenance of a polytope in the computer memory is proposed.

1. Introduction

Apart from the well-known gradient methods for convex programming, optimization methods connected with the so-called central sections have been recently widely spread. In this paper we discuss the main versions proposed up to now and give some new approaches. We shall consider not only the theoretical aspects, but the practical effeciency of the algorithms as well, and even more attention will be paid to the latter.

2. The method of Centres of Gravity (MCG)

We consider the problem of minimizing a convex function $f(x)$ over a convex compact $M_0 \subset \mathbb{R}^n$, $|M_0|>0$ ($|\cdot|$ denotes the volume). The MCG [1, 2] actually consists of the following four components.

I) Let $x_0 \in int\, M_0$. By computing the value $grad\, f(x_0)$ we can cut off a portion of M_0 which does not contain the searched minimum point $x_{\min}$. Namely, $x_{\min} \in M_1$, where M_1 is the intersection of M_0 and the half-space

$$\langle grad\ f(x_0),\quad x-x_0\rangle \leqq 0. \tag{1}$$

(Naturally, if $grad\, f(x_0)=0$, then $x_0=x_{\min}$).

In the case of non-differentiability of f at the point x_0 one could use its subgradient. But in practice this is a rare situation in view of the algorithm's nature and the well-known fact that a convex function is differentiable almost everywhere.

If we can also calculate the value of f at a given point, then by virtue of f being convex, we can change the inequality (1) by the stronger inequality

$$\langle grad\ f(x_0), x-x_0\rangle \leqq c-f(x_0), \tag{2}$$

where c is the minimal known value of f at the points of M_0 (if $f(x_0)$ is such a minimal value itself (1) and (2) coincide). We denote this stronger variant of I by $\hat{I}$.

II) Let x_0 be the center of gravity of M_0. Any hyperplane passing through x_0 divides M_0 into two parts such that the volume of each part does not exceed [3] :

$$(1-(\frac{n}{n+1})^n)|M_0| \leqq (1-e^{-1})|M_0|.$$

By combining I and II it is possible to localize the point $x_{\min}$ in the sets $M_0 \supset M_1 \supset M_2 \supset \ldots$ whose volumes are exponentially decreasing :

$$|M_k| \leqq (1-e^{-1})^k |M_0|, \quad k=1, 2,\ldots \tag{3}$$

It is clear, that this estimate holds in the case one uses $\hat{I}$, as well.

III) Though decrease in volume does not imply a decrease in diameter, using arguments relevant to convexity, one obtains a solution of the problem of ε-minimization of f, i.e. one can find a point $x_\varepsilon \in M_0$ such that

$$f(x_\varepsilon) \leqq f(x_{\min}) + \varepsilon,$$

where $\varepsilon > 0$ is arbitrarily small. Namely, suppose that m steps of the algorithm have been performed. Let x_k be the center of gravity of M_k, $k=0, 1,\ldots, m$ and x_m^* is that point of the sequence $x_0, x_1,\ldots, x_m$ at which f has the smallest value. The following holds true [4] :

$$f(x_m^*) - f(x_{\min}) \leqq (\max_{M_0} f - \min_{M_0} f)(1-e^{-1})^{m/n}. \tag{4}$$

Hence, the smallness in volume of M_k implies the proximity of $f(x_m^*)$ to the minimum of f on M_0.

IV) When k is growing, the structure of the set M_k may assume a complicated form, which causes some difficulties. To avoid this, one can enclose M_k from time to time into a "simple structured" set M_k' which has a volume as small as possible. There are many ways to obtain such an inclusion ; the important fact is that every variant keeps the exponential decrease in volume.

The MCG differs considerably from the gradient methods. In contrast to them, MCG actually needs no additional assumptions for f exept convexity. With gradient methods attention is concentrated upon a monotonic decrease of the function f and the question of localizing the point $x_{\min}$ is not considered at all. For MCG, on the contrary, the basic problem is to contract the domain of localization.

The main advantage of the method is the possibility to a priori determine the number of needed operations ; the main difficulty is the necessity to deal with domains of localization (maintenance of M_k in the computer memory, determination of centers of gravity, cut-operations, procedures of inclusion according to IV). The specific weigth of the arising difficulties is defined by several factors — the dimension of the space, the chosen version of the method and, what

is to be specially underlined — the cost of computing the values of f and $grad\, f$ at a given point. Approximately speaking, the more expensive are the latter computations the more justified is the work with domains of localization. The smallness of n does not always imply a little cost of these computations. One such example is given in [1] (the problem of block convex programming with an arbitrary number of variables and not great number n of composition constraints). The other example is the linear time-optimal control problem, where n is usually not great, but the cost of computing $f(x)$ and $grad\, f(x)$ is significant (especially when high precision is required).

Concerning IV, we would like to note, that the original idea got an interesting extension. Independently and almost simultaneously A. S. Nemirovskii and D. B. Iudin [5] and N. Z. Shor [6] proposed to use ellipsoids as localization domains M_k and to enclose on every step the newly obtained domain (half-ellipsoid) into an ellipsoid of minimal volume. Since the inclusion operation is not connected with special difficulties, this version of MCG is practically realizable for great n, too. On the other hand, the speed of volume decrease noticeably slows down and the number of operations for computing values of f and $grad\, f$ grows approximately n times. Thus, practical efficiency could be expected in problems for which these computations are not very expensive. The problem of linear programming serves as such an example. By the way, the ellipsoid method gained fame after L. G. Khachiyan [7] had used it to prove the polynomial complexity of the linear programming problem. Althogh this result undoubtedly has theoretical significance, from practical point of view non-linear programming remains to be the natural scope of implementation for any variant of MCG, since for linear programming there exists a variety of effective algorithms.

Being one of the variants of MCG (in the broad sense of the word), the ellipsoid method has nowadays a variety of modifications [see 8, 9]. One more version of MCG was proposed independently in [10] and [11]. Here simplexes were taken as localisation domains. The reduction in volume of localization domains is less regular than at the ellipsoid method. With the ellipsoid method the volumes decrease geometrically with the ratio

$$\left(1-\left(\frac{n}{n+1}\right)^n\right)^{1/n} \sim 1-\frac{1}{ne}.$$

For the volumes of simplexes such a general formula does not exist (one can give only an upper bound). The experimental data (for $n \leqq 10$) available to us show that the volumes of simplexes decrease appreciably faster than the volumes of ellipsoids.

3. Implementable versions of MCG without using inclusion

In case the computation of $f(x)$ and $grad\, f(x)$ is highly expensive, the above said implies that it is very desirable to manage without IV, avoiding in this way the increase of volumes. We consider now the possibilities, which arise here.

Let our algorithm consist of components I-III only. Suppose that the initial localization domain M_0 is a convex polytope in R^n; hence, the subsequent localization domains M_k are also convex polytopes.

The labouriousness of the operation for determining the centers of gravity of M_k grows rapidly with the growth of n: for $n=2$ it is not great, for $n=3$ it is

essentially greater, with further growth of the dimension the size of the required work quickly leaves the real limits.

The basic way to lower the labouriousness of the method is to change the centers of gravity x_k by some kind of points x'_k, which are "central" enough and comparatively easy to be obtained. We shall consider three approaches of this sort.

3.1. Convex combination of vertices. One of the natural pretenders for a "central" point of the polytope M is the center of gravity of its vertices v_1, $v_2, \ldots, v_m$:

$$z(M)=\frac{1}{m}\sum_{i=1}^{m} v_i$$

or its stochastic image

$$z(M,\ \alpha)=\sum_{i=1}^{m} \alpha_i v_i,$$

where $\alpha=(\alpha_1,\ \alpha_2, \ldots, \alpha_m)$ is random sampled point uniformly distributed in the simplex $\Sigma_{i=1}^{m} \alpha_i=1,\ \alpha_i \geqq 0,\ i=1,2,\ldots,m$.

If we know the vertices of M, we can easily obtain $z(M)$, respectively $z(M, \alpha)$. Thus, the computational work is reduced to passing from the vertices of M_k to the vertices of M_{k+1} in an effective way. It is of great importance to rationally organize the corresponding data base. As far as this problem is of independent interest, we consider it in detail in section 5. We mention now only the fact that we have to preserve all the m_k vertices of the polytope M_k. Therefore, it is important how m_k varies. One might expect that along with k, m_k will also increase. But, as the experiments show, this does not happen and the maximal value of m_k, as a rule, allows MCG to be successfully used for $n \leqq 10$. Another strongpoint of the considered version is the quick decrease in volume. Unfortunately, both these advantages are verified only empirically. It is impossible to give a guaranteed estimate for volume contraction (for $n>2$) as it is shown by the example of pyramid with great number of vertices in the base and section parallel to the base. Such situations are very rare in practice. One of the aims of the randomized version connected with $z(M, \alpha)$ is, roughly speaking, to lower the probability for appearance of such situations. Another aim is to dispose with a flexible scheme, which could be stochastically varied (at the expense of randomness of α). An indirect justification for different stochastic versions of the method is the result [12].

3.2. Uniformly distributed representatives. Suppose that we have an easy way to generate random uniformly distributed points in the polytope M_k. One of the methods we can apply is the method of "random walking on chords" [13]. It is clear, that if we have $l \gg 1$ independent uniformly distributed in the volume of M_k points, then their mean value is close to the center of gravity of M_k in probability sense (with the growth of l the standard error is of order $l^{-1/2}$). Hence, the

question is to obtain such points effectively enough. Let us call them representatives of M_k.

Let l be some fixed number of representatives. Now M_k will be characterized by two "constituents" : system of linear inequalities and system of representatives. We consider the transition from M_k to M_{k+1}.

Let z_k be the arithmetic mean of the representatives of M_k and $f(z_k^*)$ be the smallest value ("the record") of f among the values $f(z_0), f(z_1), \ldots, f(z_n)$. According to $\hat{I}$ the system of linear inequalities for M_k contains (apart from the initial group of inequalities defining M_0) inequalities of the kind :

$$\langle grad\, f(z_i), x - z_i \rangle \leqq f(z_{k-1}^*) - f(z_i), \quad i = 0, 1, \ldots, k-1. \tag{5}$$

The transition to the system of linear inequalities defining M_{k+1} is carried out in the following way. If $f(z_k) < f(z_{k-1}^*)$, the righthand side of the inequalities (5) is to be reduced, i. e. $f(z_{k-1}^*)$ is to be changed by $f(z_k^*)$ $(= f(z_k))$. No matter whether we had to recalculate the right-hand sides of system (5) or not, we add one new inequality :

$$\langle grad\ f(z_k), x - z_k \rangle \leqq f(z_k^*) - f(z_k).$$

It is by far not so simple to solve the question of representatives of M_{k+1}. It is easy to check which of the representatives of M_k remain in M_{k+1} ; let us denote them by l' $(< l)$. We are to complement the number of representatives of M_{k+1} with respect to l. It is easy to show, that in the case of "random walking on chords" the number of operations needed in order to come over from M_k to M_{k+1} is proportional to $nkml$. As far as n, m and l are fixed, the labouriousness of a transition from step to step grows linearly with respect to k, while the volumes of M_k decrease exponentially. Thus, the increase of labouriousness of a step is not so important.

3.3. Combination of MCG and gradient methods. As it has been mentioned earlier the MCG essentially differs from gradient methods. This does not mean, however, that these two approaches are mutually exclusive and, what is more, their combination can be very expedient in practice. We shall consider two examples of such combinations.

In the first place, the scheme of the algorithm can be the following. At the beginning one can use the MCG (in some of the versions presented above) until, for instance, the maintenance of the information about the subsequent polytope M_k becomes too hard. Usually in this moment the volume of M_k turns out to be small enough and hence the record value $f(x_k^*)$ is close enough to the minimum. Therefore, it is natural to choose x_k^* as an initial point for some kind of gradient methods. Thus, here the MCG is used in order to obtain a qualitative initial approximation, which as it is well-known, is of great importance for the majority of gradient methods.

Secondly, one can proceed as follows. Suppose k steps have been carried out using MCG with the component $\hat{I}$. Let x_k' and x_k'' be the intersection points of

M_{k+1} with the ray starting from the point x_k and going into direction of $-grad f(x_k)$ (we do not consider here the possibility of the ray not intersecting M_{k+1}). It is clear, that if x_k is a record point (i.e. $x_k^* = x_k$), then either $x_k = x_k'$ or $x_k = x_k''$. As a "central" point x_{k+1} we chose some point in the interval (x_k', x_k'') (for instance, $(x_k' + x_k'')/2$). The sense of this recommendation is obvious enough. Such a procedure could be considered as a descent in the direction of the antigradient with the length of the step defined by the information about the localization domain. On the other hand, going into the anti-gradient direction we enlarge the chance to obtain a next record value, which, in connection with $\hat{I}$, can imply a sharp reduction in the volume of M_{k+2}. It is clear, that the case when the point x_k is a record itself, is especially favourable. Therefore, it is proper to include the proposed approach into every version of MCG (which does not use IV) and apply it on the steps, where x_k turns out to be a record point.

4. MCG for quasi-convex functions

Until now we assumed that the function to be minimized is convex. However, the basic idea of the algorithm is applicable for quasi-convex functions, too. Recall that f is called quasi-convex over a convex set $M_0 \subset R^n$ if

$$f(\lambda x_1 + (1-\lambda)x_2) \leqq \max\{f(x_1),\ f(x_2)\}$$

for all $\lambda \in [0,\ 1]$ and all $x_1,\ x_2 \in M_0$.

For the sake of presentational simplicity we confine ourselves to a continuous differentiable function f (although, the differentiability, as in the convex case, in fact, is unessential). We suppose as well, that *grad f* vanishes only at the minimum points of f.

By using the already employed enumeration of the components of the method, we can briefly circumscribe the case in a following way. The components I-IV remain valid, although the justification of III needs modification ; $\hat{I}$ does not hold. Indeed, (1) easily follows from the quasi-convexity of f. The components II and IV have pure geometrical character and are not in connection with the form of the function. Let us consider in detail the component III.

The inequalities (4) for a quasi-convex function, generally speaking, are not true. In order to realize the transition from smallness in volume to a smallness in error with respect to the functional, we will use some additional information — the Lipschitz constant L and the diameter d of M_0. We will show that, instead of (4), the following inequalities hold :

$$f(x_m^*) - f(x_{\min}) \leqq L d (1 - e^{-1})^{m/n}. \tag{6}$$

In fact, let $x_{\min}$ be any minimum point of f over M_0 (obviously, $x_{\min} \in M_m$), x' be the nearest to $x_{\min}$ point of the closure of $M_0 \setminus M_m$, $\rho = \| x_{\min} - x' \|$. The value of f at a record point x_m^* is not greater than the value at any point from the closure of

$M_0 \setminus M_m$ (for the quasi-convex case it can be proved in the same way as for the convex one). Therefore

$$f(x_m^*) - f(x_{\min}) \leqq f(x') - f(x_{\min}) \leqq L\rho. \tag{7}$$

Furthermore, every interval $(x_{\min}, y)$, $y \in M_0 \setminus M_m$, is not greater in length than d and contains an interval $(x_{\min}, y')$ with length not less than ρ. Hence

$$\frac{|M_m|}{|M_0|} \geqq (\rho/d)^n. \tag{8}$$

By comparing (7) and (8), we obtain

$$f(x_m^*) - f(x_{\min}) \leqq L d \left(\frac{|M_m|}{|M_0|}\right)^{1/n},$$

which in view of (3) gives the inequality (6).

It can occur that the Lipschitz constant of f is unknown or does not exist, but the estimate for the continuity module of f

$$|f(x+y) - f(x)| \leqq g(\|y\|), \quad (g(t) \to 0 \text{ for } t \to 0),$$

is available. The same arguments show that then

$$f(x_m^*) - f(x_{\min}) \leqq g(1 - e^{-1})^{m/n} d).$$

Thus, if $g(t) \leqq ct^{\varepsilon}$ for some $\varepsilon > 0$, the exponential convergence with respect to the functional is preserved.

We note, that the estimate (6) can be obtained with the help of D. B. Iudin's result [14], which is formulated in other terms.

We dealt with the case of quasi-convex functions in view of the fact that they are very important in the applications. As an example we cite the already mentioned linear time-optimal control problem. This problem can be reduced to the problem of minimizing some quasi-convex function f. The function f is implicitly described, but it is possible to calculate the values of f and *grad* f at a given point. The calculations are very expensive even for not large problems. Apparently, the MCG is the most effective way to solve this problem. We are not going to deal in detail with this problem in the present paper.

5. Maintenance of a polytope in the computer memory

As it was mentioned above, the rational organization of a data base when dealing with polytopes is of great importance. We propose now an effective way to maintain and update a polytope.

Any polytope can be represented as a set of linear inequalities or, what is the same, as a convex hull of its vertices. The main operations, we want to perform effectively, are the following :

1. Check if a point lies in the polytope.
2. Check if a new inequality intersects the polytope, is "superflous" or "anti-coincident".
3. Add a new inequality, i.e. form a new (smaller) polytope.
4. Create a (random) point in the polytope using its vertices.

In order to perform operations 1. and 4., we need a set of inequalities defining the polytope and a set of polytope's vertices. Moreover, having in mind operations 2. and 3., these two sets should be properly connected.

The base of the structure we suggest consists of the so-called lists. Every list represent an inequality or a vertex and consists of the triple :

— real vector of fixed size $n+1$;
— non-negative integer — the length of the list ;
— a set of different integers sorted in an ascending order.

The first part of the list contains the corresponding coordinates (of a vertex or of a normal vector and the right-hand side defining the inequality). The integers in the last part of the list are pointers to some other lists. A list is called "short" when its third part is missing and "full" otherwise. Actually, each vertex of the polytope is itself a full list consisting of faces incident with this vertex and each face of the polytope is a list of vertices incident with this face (generally a short one).

At the very beginning we create a common pool of empty lists in the computer memory. Every list is defined by its starting address. The main operations over the lists are the following :

A. Get a list from the pool.
B. Return a list to the pool.
C. Insert an integer into a list.
D. Delete an integer from a list.
E. Compare two full lists and in case the number of common elements is not less than a given m, create a list containing the common elements.

The data structure we propose is built according to the concept of lists described above. The key procedure in the structure is the operation 3. It can be carried out in the following steps :

1°. Separate the list of all vertices into a set of "old" vertices (i.e. satisfying the new inequality) and "marked" vertices (i.e. not satisfying the new inequality).
2°. Delete the marked vertices from all lists.
3°. Delete faces with empty lists of vertices.
4°. Create an empty list of "new" vertices for the face "defined" by the new inequality.
5°. Choose convenient pairs of old and marked vertices and create a new vertex (together with its list) between them.
6°. Clear marked vertices from the computer memory.
7°. Add the new vertices to corresponding lists.
8°. Create the set of all vertices as an union of the sets of old and new vertices.

Let us note that it is possible to use the dual representation of the polytope exchanging the vertices and the faces in the structure considered above. Then operation 3. is an adding of a new vertex to the description of the polytope.

FORTRAN programs were written by the authors for the presented structure and are available to users. The programs have been tested with ES-1040 and IBM-PC computers. The efficiency of the data structure depends heavily on the number of polytope vertices and faces. For random n-polytopes ($n \leqq 10$) it turns out that the amount of memory needed is not very large.

An additional advantage of the structure is that it allows implementing the original MCG (when full lists are applied) by splitting the polytope into simplexes. This problem will not be considered in detail here.

6. Conclusion

From all stated above it follows that the MCG represents a whole family of algorithms joined by some common ideas. We outlined a part of the existing versions only. The choice of a concrete variant depends on many considerations – the dimension of the problem, the information available for the function f, the cost of computing the values of f and *grad* f at a given point, etc.

We did not aim to give a survey on all the newest results concerning MCG. Our aim was, mainly, to draw attention to the practical applicability and efficiency of some versions of the method, which apply neither inclusion of the localization domains into "simple" domains (which is volume-increasing) nor exact computing of centers of gravity. It seems to us that this implementability and efficiency have not been entirely realized by operations researchers up to now.

Many questions arise in connection with the problems considered. The most interesting of them are, in our opinion, in the intersection of probability theory and geometry of convex polytopes. The answers to these questions can be obtained both in the way of theoretical investigations and with the help of large-scaled computer experiments.

References

1. A. Iu. Levin. On an Algorithm for the Minimization of Convex Functions. *Doklady Akademii Nauk SSSR*, **160**, 1965, 1244-1247. (Russian).
2. D. J. Newman. Location of the Maximum on Unimodal Surfaces. *J. ACM*, **12**, 1965, 395-398.
3. B. Grunbaum. Partitions of Mass-destributions and of Convex Bodies by Hyperplanes. *Pacific. J. Math.*, **10**, 1960, 1257-1261.
4. A. S. Nemirovskii and D. B. Iudin. Complexity of Problems and Efficiency of Optimization Methods. *Nauka, Moscow*, 1979. (Russian).
5. D. B. Iudin and A. S. Nemirovskii. Informational Complexity and Effective Methods of Solution for Convex Extremal Problems. *Ekonomika i Math. Metody*, **12**, 1976, 357-369. (Russian).
6. N. Z. Shor. Cut-off Method with Space Extension in Convex Programming Problems. *Kibernetika*, **1**, 1977, 94-95. (Russian).
7. L. G. Khachiyan. A Polynomial Algorithm in Linear Programming. *Doklady Akademii Nauk SSSR*, **244**, 1979, 1093-1096. (Russian).
8. R. G. Bland, D. Goldfarb and M. J. Todd. The Ellipsoid Method : A Survey. *Operations Research*, **29(6)**, 1981, 1039-1092.
9. I. L. Goffin. Variable Metric Relaxation Methods, Part II, The Ellipsoid Method. *Math. Programm.*, **30(2)**, 1984, 147-162.
10. B. Yamnitsky and L. A. Levin. An Old Linear Programming Algorithm Runs in Polynomial Time. – In: 23rd Annu. Symp. Found. Comput. Sci., Chicago, III, 3-5 Nov., 1982, *Silver spring, Md*, 1982, 327-328.
11. L. T. Ashchepkov, B. I. Belov, V. P. Bulatov and others. Methods for Solving Mathematical Programming and Optimal Control Problems. Nauka, Novosibirsk, 1984. (Russian).
12. A. Iu. Levin and A. S. Shvarc. On a Scheme of a Random Search. Proceedings of Seminar on Functional Analysis. 7, Voronezh, 1963, 67-69. (Russian).
13. Iu. V. Rusin. On a Markov Generation of Uniform Distribution in a Multidimensional Domain. – In : Heuristic Algorithms in Optimization. Yaroslavl, 1981, 79-82. (Russian).
14. D. B. Iudin. Mathematical Programming in Order Scales. *Izvestiia Akademii Nauk SSSR, Tekhnicheskaia Kibernetika*, **2**, 3-17. (Russian).

Institute of Mathematics
Bulgarian Academy of Sciences
1090 Sofia, BULGARIA

Yaroslavl State University
ul. Sovietskaia d. 14
150000 Yaroslavl, USSR

Received 20.07.1987