



On minimization of a quadratic function with one negative eigenvalue

Ilya Minarchenko¹ · Oleg Khamisov¹

Received: 21 February 2020 / Accepted: 6 October 2020 / Published online: 12 October 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

It is well known that a quadratic programming minimization problem with one negative eigenvalue is NP-hard. However, in practice one may expect such problems being not so difficult to solve. We suggest to make a single partition of the feasible set in a concave variable only so that a convex approximation of the objective function upon every partition set has an acceptable error. Minimizing convex approximations on partition sets provides an approximate solution of the nonconvex quadratic program that we consider. These minimization problems are to be solved concurrently by parallel computing. An estimation of the number of partition sets is given. The study presents a computational comparison with a standard branch-and-bound procedure.

Keywords Quadratic programming · Global optimization · Branch-and-bound method · Parallel computing

1 Introduction

Quadratic nonconvex programming has a wide range of applications and is an intensively studied field of global optimization. The number of publications devoted to this topic is quite impressive, see, for example, [1–3] and references therein. The main issue of the paper is connected to the result published in [4]. Namely, nonconvex quadratic programming problem with one negative eigenvalue in the objective is NP-hard. Properties of a quadratic function with one negative eigenvalue and the

This study was funded by Russian Foundation for Basic Research according to the research Project No. 18-07-01432.

✉ Oleg Khamisov
khamisov@isem.irk.ru
Ilya Minarchenko
minar@isem.irk.ru

¹ Melentiev Energy Systems Institute of Siberian Branch of the Russian Academy of Sciences, Lermontov st. 130, Irkutsk, Russia 664033

corresponding theoretical results can be found in [5,6]. We get interested in how difficult on average the problem of global minimization of quadratic objective is with exactly one negative eigenvalue. We assume that the eigenvalues and eigenvectors of the matrix in the minimized function are known. Testing the problem from [4] showed that NP-hardness follows from exponential growth of the size of the feasible set. If the size is bounded then the problem can be solved in a reasonable time. The paper investigates this topic in details.

2 Problem formulation

We consider a quadratic programming problem

$$f(x) = x^T Q x + x^T c \rightarrow \min_x, \quad (1)$$

$$x \in X = \{x \in \mathbb{R}^n \mid Ax \leq b\}, \quad (2)$$

where Q is an indefinite $n \times n$ matrix with exactly one negative eigenvalue, A is an $m \times n$ matrix. The feasible set X is assumed to be nonempty and bounded. Here x^T denotes the transpose of vector x .

Let us represent the matrix Q such that

$$Q = W \Lambda W^T, \quad \Lambda = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_n\}, \quad (3)$$

where $\lambda_1, \lambda_2, \dots, \lambda_n$ are the eigenvalues of the matrix Q rearranged by increasing order, i.e. $\lambda_1 \leq \dots \leq \lambda_n, \lambda_1 < 0, \lambda_i \geq 0, i = 2, \dots, n$; W is a matrix, which columns are the eigenvectors of Q sorted according to its eigenvalues. Λ is a diagonal matrix with $\lambda_1, \lambda_2, \dots, \lambda_n$ as the diagonal elements. It is well known [7] that $W^{-1} = W^T$. The linear transformation

$$y = W^T x \quad (4)$$

reduces the problem (1), (2) to the separable form, thus we get the following problem with the separable objective function:

$$g(y) = \sum_{i=1}^n \lambda_i y_i^2 + y^T d \rightarrow \min_y, \quad (5)$$

$$y \in Y = \{y \in \mathbb{R}^n \mid Dy \leq b\}, \quad (6)$$

where $d = W^T c$ and $D = AW$. The problems (1), (2), and (5), (6) are equivalent in the sense that x^* is a solution of (1), (2) iff $y^* = W^T x^*$ solves (5), (6). Moreover values of the objectives also coincide. The further discussion refers mainly to the latter optimization statement since it is more convenient for analysis.

3 Minimization procedure

The objective function (5) is concave in y_1 and is convex in other variables. Therefore the main idea of the minimization procedure for problem (5), (6) is a partition of the feasible set Y in variable y_1 only. Let

$$\tilde{Y} = \{y_1 \in \mathbb{R}^1 \mid \alpha \leq y_1 \leq \beta\}$$

be a projection of the set Y onto the y_1 -axis. Divide the set \tilde{Y} into p intervals:

$$\tilde{Y} = \bigcup_{i=1}^p \{y_1 \in \mathbb{R}^1 \mid \alpha_i \leq y_1 \leq \beta_i\}, \tag{7}$$

where $\alpha \leq \alpha_i < \beta_i \leq \beta$, $\beta_i = \alpha_{i+1}$ for $i = 1, \dots, p - 1$, and $\alpha_1 = \alpha, \beta_p = \beta$. Then we define a partition $\{Y_1, Y_2, \dots, Y_p\}$ of Y as follows:

$$Y = \bigcup_{i=1}^p Y_i, \quad Y_i = \{y \in Y \mid \alpha_i \leq y_1 \leq \beta_i\}, \quad i = 1, 2, \dots, p. \tag{8}$$

For every partition set Y_i we underestimate the concave part $\lambda_1 y_1^2$ of the objective by an affine function l_i such that $\lambda_1 y_1^2$ and l_i intersect at points α_i and β_i :

$$\lambda_1 y_1^2 \geq l_i(y_1) = \lambda_1(\alpha_i + \beta_i)y_1 - \lambda_1\alpha_i\beta_i, \quad \alpha_i \leq y_1 \leq \beta_i. \tag{9}$$

From (9) we obtain a convex approximation of g :

$$g(y) \geq \varphi_i(y) = l_i(y_1) + \sum_{i=2}^n \lambda_i y_i^2 + y^T d, \quad y \in Y_i, \quad i = 1, 2, \dots, p.$$

This inequality can be presented in the following equivalent form:

$$g(y) \geq \varphi(y) = \min_{1 \leq i \leq p} l_i(y_1) + \sum_{i=2}^n \lambda_i y_i^2 + y^T d, \quad y \in Y.$$

It is known [8] that

$$g(y) - \varphi_i(y) \leq -\frac{1}{4}\lambda_1(\alpha_i - \beta_i)^2, \quad y \in Y_i,$$

which results in the following estimation of the approximation error upon the entire feasible set:

$$g(y) - \varphi(y) \leq -\frac{1}{4}\lambda_1 \max_{1 \leq i \leq p} \{(\alpha_i - \beta_i)^2\}, \quad y \in Y. \tag{10}$$

Theorem 1 For a given $\varepsilon > 0$, let the partition (7) satisfy

$$\beta_i - \alpha_i \leq 2\sqrt{\frac{\varepsilon}{|\lambda_1|}}, \quad i = 1, 2, \dots, p, \quad (11)$$

and y^* be a minimum point of φ upon Y . Then the following inequality holds:

$$g(y^*) \leq g(y) + \varepsilon, \quad y \in Y. \quad (12)$$

Proof The inequalities (11) are equivalent to

$$\varepsilon \geq -\frac{1}{4}\lambda_1 \max_{1 \leq i \leq p} \{(\alpha_i - \beta_i)^2\}. \quad (13)$$

As $\varphi(y^*) \leq \varphi(y)$ for every $y \in Y$, (12) directly follows from (10) and (13). \square

From (10) to (13) we have that for every given tolerance $\varepsilon > 0$ we may define a partition of Y with sufficiently small widths of intervals in y_1 , so that

$$g(y) - \varphi(y) \leq \varepsilon, \quad y \in Y. \quad (14)$$

Underestimation of g by φ in that manner allows us to perform the partition (8) only once such that the inequality (14) holds. By Theorem 1, every minimizer of φ upon Y is an approximate solution of (5), (6). In order to minimize φ , we solve p convex quadratic programming problems

$$\varphi_i(y) \rightarrow \min, \quad y \in Y_i, \quad i = 1, 2, \dots, p. \quad (15)$$

Let y^{*i} be a solution of the i th problem in (15). Then

$$\min \varphi(y) = \min_{1 \leq i \leq p} \varphi_i(y^{*i}),$$

and a minimum point of φ is the corresponding vector among y^{*1}, \dots, y^{*p} . The optimization problems (15) are suggested to solve concurrently by parallel computing, since they are independent of each other.

As every partition set implies solving a convex quadratic program, the number of partition intervals p has a significant impact on how fast the set of problems (15) would be solved. For a particular problem (5), (6), one can compute the value of p satisfying (11) by

$$p = \left\lceil \frac{1}{2}(\beta - \alpha) \sqrt{\frac{|\lambda_1|}{\varepsilon}} \right\rceil, \quad (16)$$

where $\lceil a \rceil$ is the least integer greater than or equal to a number a . Additionally, an upper estimation of p needed to meet (11) may be established. Since X is bounded,

there exists a box constrained set $\widehat{X} = \{x \in \mathbb{R}^n \mid x^{\min} \leq x \leq x^{\max}\}$ such that $X \subseteq \widehat{X}$. It is better to choose \widehat{X} so that it has the minimal diameter.

Theorem 2 For a given $\varepsilon > 0$, there exists a positive integer p satisfying

$$p \leq \left\lceil \frac{1}{2} \sqrt{\frac{n|\lambda_1|}{\varepsilon}} \max_{1 \leq i \leq n} \{x_i^{\max} - x_i^{\min}\} \right\rceil \tag{17}$$

such that the partition (7) meets (11).

Proof Inscribe \widehat{X} in a ball $B = \{x \in \mathbb{R}^n \mid \|x - x_0\| \leq r\}$, where $x_0 = \frac{1}{2}(x^{\min} + x^{\max})$, $r = \|x^{\max} - x_0\|$, and $\|\cdot\|$ is the 2-norm. As $W^T W = I$, where I is the identity matrix, the projection of B onto the y_1 -axis after the transformation (4) has the length equal to the diameter $2r$ of B . Since $X \subset B$, there exist an integer p and bounds (α_i, β_i) , $i = 1, \dots, p$, such that (11) holds and $p \leq \lceil 2r/l \rceil$, where $l = 2\sqrt{\varepsilon/|\lambda_1|}$. The latter inequality directly implies (17). \square

It follows from Theorem 2 that, if $X \subseteq \widehat{X}$ and $\lambda_1, x^{\min}, x^{\max}$ do not depend on n , then the number of intervals $p(n)$ has growth rate \sqrt{n} . In the next section we consider an example, where the first component of x^{\max} grows rapidly as n increases.

4 Computational experiment

The proposed method with a single partition was computationally compared with a standard branch-and-bound procedure [9]. Let denote the former method by SP and the latter one by BB. In BB algorithm, partitioning is performed by bisection and, additionally, partition sets that certainly do not admit a solution are excluded from the consideration. More precisely, a partition set should be neglected if its lower bound of the objective exceeds the best known objective value (the record). For a particular partition set Y_i in BB procedure, lower bound is a minimum of φ_i on Y_i .

We consider three groups of test problems. For every group, the result of the computation for both SP and BB algorithms is presented in a separate table. The names of columns are as follows. n is the number of variables, m is the number of linear constraints in (2) excluding box constraints, SP Int. is the number of intervals p computed by (16), SP Th.Int. is the theoretical estimation of p computed by (17), SP Time is the time of SP algorithm in seconds, BB Int. is the number of convex problems solved in BB algorithm, BB Time is the time of BB algorithm in seconds, Rel.Err. stands for an actual relative error computed after the solution is obtained. In all tables, every row corresponds to a single randomly generated problem. All random values were chosen from specific intervals according to a uniform distribution. To compute the right hand side of (17), we took box constraints of the problems' definitions as the set \widehat{X} .

The program was implemented in AIMMS 4.66 modelling environment [10], problems (15) in both methods were solved by IBM ILOG CPLEX 12.9 solver [11]. In the SP method, we solved problems (15) concurrently in parallel using AIMMS GMP library facilities and, additionally, the number of threads used by one CPLEX session was limited to 1. In BB, problems of type (15) were solved sequentially, however the

number of threads used by one CPLEX session was not limited so that every convex problem in BB was solved by multiple threads. We used default values for all other CPLEX parameters. Computation was performed on a computer with AMD Ryzen 7 1700X @ 3.4GHz CPU (8 cores, 16 threads), 16 Gb RAM. Preliminary experiments showed that the time of SP is minimal when the method uses 8 threads on this computer. Hence, for the SP algorithm, the number of simultaneous solution of convex problems was limited to 8.

Example 1 In problem (1), (2), set $c = -(Q - \lambda_1 I)x^*$, $x^* = (1, -1, \dots, 1, -1)^T$, $X = \{x \in \mathbb{R}^n \mid -1 \leq x \leq 1\}$. Vector x^* is a solution of (1), (2) [12]. Values of λ_1 , λ_i , $i = 2, \dots, n$, and of elements of W were randomly chosen from intervals $[-11, -7]$, $[0, 20]$, $[-10, 10]$ respectively. Eigenvectors of Q were computed by applying the Gram–Schmidt process to columns of W . Then we defined Q by (3). For these problems, we set $\varepsilon = 10^{-2}$, because it provided an acceptable relative error in a solution point. The computational results of SP and BB methods are in Table 1. We did not apply the SP algorithm to problems with $n > 500$ due to considerable time of computation, more than 10 minutes per one problem. In all instances both algorithms found the solution x^* . As there are box constraints only, $m = 0$ for all problems. The number of intervals turns out to be close to its theoretical estimation (17) and grows in accordance with it as n increases. On the other hand, in the BB algorithm, the number of convex problems solved by the method is small and is approximately constant for all dimensions we considered. The “bottleneck” in this case is the time CPLEX spends for solving one quadratic programming problem.

Table 1 Results of experiment for problems with known solution (Example 1)

n	m	SP Int.	SP Th.Int.	SP time	BB Int.	BB time	Rel.Err.
50	0	192	226	0.710	17	0.610	0.000006
100	0	286	331	1.760	19	1.160	0.000003
150	0	341	398	5.460	17	1.490	0.000002
200	0	345	402	14.580	19	2.750	0.000002
250	0	440	505	40.410	19	4.390	0.000001
300	0	495	574	81.260	19	5.980	0.000001
350	0	452	522	129.420	19	8.320	0.000001
400	0	490	560	277.540	19	12.280	0.000001
500	0	547	627	737.040	17	18.830	0.000000
600	0	593	687	–	19	32.960	0.000000
700	0	662	759	–	19	53.760	0.000000
800	0	676	773	–	21	87.570	0.000000
900	0	837	966	–	21	120.500	0.000000
1000	0	756	873	–	21	157.350	0.000000
1250	0	995	1154	–	17	252.900	0.000000
1500	0	1062	1228	–	23	606.840	0.000000

Example 2 In this case of (1), (2) all parameters of the objective function were generated randomly in the same way as in Example 1, except for vector c , the elements of which were randomly chosen from $[-30, 30]$. Elements of A were generated from $[-15, 15]$. Vector b was computed such that every constraint in (2) defined a tangent hyperplane to a ball with the radius 4 and random center, the elements of which were from $[-5, 5]$. In order to guarantee the boundedness, we added box constraints $-10 \leq x \leq 10$. We set $\varepsilon = 10^{-2}$. Table 2 contains the result of experiment. It shows that the number of intervals in SP is significantly larger than in Example 1. The reason for it is that variables have a larger bounds so that the length of projection \tilde{Y} increases. In spite of the promising estimation (17), large number of intervals together with a significant time of solving convex QP leads to considerable time of the SP method. The number of convex problems solved in BB slightly exceeds that in Example 1, but still is nearly constant as n increases. The number m of non-box linear constraints increases the time CPLEX needs to solve a QP problem. One can observe this effect if we compare the columns BB Int. and BB Time for problems with the same n . For example, the problem with $n = 1000$, $m = 1500$ takes more than 1.5 times of com-

Table 2 Results of experiment for random problems (Example 2)

n	m	SP Int.	SP Th.Int.	SP time	BB Int.	BB time	Rel.Err.
100	100	1802	2750	27.950	31	4.020	0.000001
100	200	1363	2836	33.620	27	4.570	0.000002
100	300	1151	2772	43.330	33	6.860	0.000002
200	200	2818	4133	288.700	31	10.400	0.000000
200	300	2726	4446	403.210	31	12.440	0.000000
200	400	2126	4238	436.570	23	10.810	0.000000
300	100	3748	4838	–	23	11.670	0.000000
300	200	3458	4804	–	25	16.360	0.000000
300	300	3260	4791	–	31	21.900	0.000000
300	600	2846	5280	–	29	32.110	0.000000
400	200	4588	5947	–	29	30.450	0.000000
400	400	4071	6116	–	25	33.790	0.000000
400	600	3623	6265	–	23	41.300	0.000000
500	300	4884	6634	–	31	59.660	0.000000
500	500	4066	6197	–	27	66.910	0.000000
500	800	3967	6683	–	25	94.390	0.000000
800	500	6090	8410	–	33	255.320	0.000000
800	800	5333	8075	–	29	302.040	0.000000
800	1000	5245	8362	–	35	400.130	0.000000
1000	800	6720	9535	–	31	525.550	0.000000
1000	1000	6203	9381	–	29	544.610	0.000000
1000	1500	5728	9742	–	29	853.340	0.000000

Table 3 Results of experiment for the maximum clique problem (Example 3)

$ V $	$ E $	n	m	SP Int.	SP Th.Int.	BB Int.	BB time	Rel.Err.
3	3	8	5	4200	4200	31	0.040	0.000000
4	3	12	11	13,600	17,000	59	0.080	0.000029
5	4	17	18	54,600	68,200	69	0.100	0.000000
6	11	23	21	268,800	273,000	197	0.300	0.000016
7	12	30	32	1,027,200	1,092,200	313	0.520	0.000030
8	14	38	44	4,355,200	4,369,000	497	0.960	0.000000
9	23	47	51	16,588,800	17,476,200	639	1.220	0.000023
10	30	57	62	52,688,000	69,905,000	1361	3.020	0.000010
11	34	68	78	262,196,000	279,620,200	1227	2.900	0.000024
12	33	80	101	1,048,579,400	1,118,481,000	2291	5.720	0.000008

putation time compared with the problem with the same n and $m = 800$. For the SP algorithm, the similar result takes place.

Example 3 As stated in [4], a specifically defined QP problem with one negative eigenvalue is equivalent to the maximum clique problem. For the graph with the set of vertices $V = \{1, \dots, v\}$ and the set of edges E this QP is

$$\begin{aligned}
 -w^2 + z - (x_1 + \dots + x_v) &\rightarrow \min_{(w,z,x,y)}, \quad 0 \leq x \leq 1, \quad y \geq 0, \\
 w &= bx_1 + b^2x_2 + \dots + b^vx_v, \\
 z &= b^2x_1 + b^4x_2 + \dots + b^{2v}x_v + \sum_{1 \leq i < j \leq v} 2b^{i+j}y_{ij}, \\
 y_{ij} &\geq x_i + x_j - 1, \quad 1 \leq i < j \leq v, \quad x_i + x_j \leq 1, \quad (i, j) \notin E, \quad (18)
 \end{aligned}$$

where $b = 4$. Without loss of generality, we add the constraint $y \leq 1$ to provide the boundedness. The key feature that makes this problem difficult to solve is that the length of projection \tilde{Y} onto w -axis (assuming $y_1 = w$) rapidly grows as v increases. Indeed, due to (18) this length may be estimated by $b + b^2 + \dots + b^v$. In spite of this, we made some experiments to illustrate how behaviour of the methods differs from the previous two examples. We assume $\varepsilon = 10^{-4}$. The set E was generated randomly in the form of a symmetric adjacency (0, 1)-matrix with zeros on its diagonal. We did not apply the SP algorithm due to a very large number of partition intervals in almost all problems we considered (see the column SP Int. in Table 3). This number exceeds 10 000 even for the graph with 4 vertices. The number of convex problems in the BB procedure also increases fast. It is easy to see that problems (15) are linear for this statement, however large numbers in coefficients raise numerical issues with increasing of v .

5 Conclusion

We consider a quadratic programming problem with one negative eigenvalue. In general, it is NP-hard. However, if input parameters such as the negative eigenvalue and the feasible set do not depend on the number of variables, then a solution can be found by solving an acceptable number of convex quadratic programming problems. The experiment shows that, for the standard branch-and-bound procedure, the number of these convex problems is nearly constant for different dimensions.

Acknowledgements We are grateful to two anonymous referees whose comments and suggestions helped us to essentially improve the paper.

References

1. Bomze, I.: Global optimization: a quadratic programming perspective. In: Di Pillo, G., Schoen, F. (eds.) *Nonlinear Optimization. Lecture Notes in Mathematics*, vol. 1989. Springer, Berlin (2010)
2. Murti, K.G.: *Optimization for Decision Making: Linear and Quadratic Models*. Springer, New York (2010)
3. Tuy, H.: *Convex Analysis and Global Optimization*. Springer, Berlin (2016)
4. Pardalos, P.M., Vavasis, S.A.: Quadratic programming with one negative eigenvalue is NP-hard. *J. Glob. Optim.* **1**, 15–22 (1991)
5. Cambini, A., Martein, L.: *Generalized Convexity and Optimization: Theory and Applications*. Springer, Berlin (2009)
6. Mishra, S.K., Giorgi, G.: *Invexity and Optimization*. Springer, Berlin (2008)
7. Jeffrey, A.: *Matrix Operations for Engineers and Scientists: An Essential Guide in Linear Algebra*. Springer, Cham (2010)
8. Pardalos, P.M., Rosen, J.B.: *Constrained Global Optimization: Algorithms and Applications*. Springer, Berlin (1987)
9. Horst, R., Tuy, H.: *Global Optimization: Deterministic Approaches*. Springer, Berlin (1996)
10. <https://www.aimms.com>. Accessed 24 July 2019
11. <https://www.ibm.com/products/ilog-cplex-optimization-studio>. Accessed 24 July 2019
12. Pardalos, P.M.: Construction of test problems in quadratic bivalent programming. *ACM Trans. Math. Softw.* **17**, 74–87 (1991)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.