r-ALGORITHM FOR MINIMIZATION OF CONVEX FUNCTIONS WITH LINEAR EQUALITIES

Anastasiia Storozhenko Petro Stetsyuk

V.M. Glushkov Institute of Cybernetics, National Academy of Sciences of Ukraine, Kyiv, Ukraine

Mathematics Days at Moldova State University 2026
Workshop Differential Equations and Operations Research
August 23–25, 2026, Chisinau, Moldova

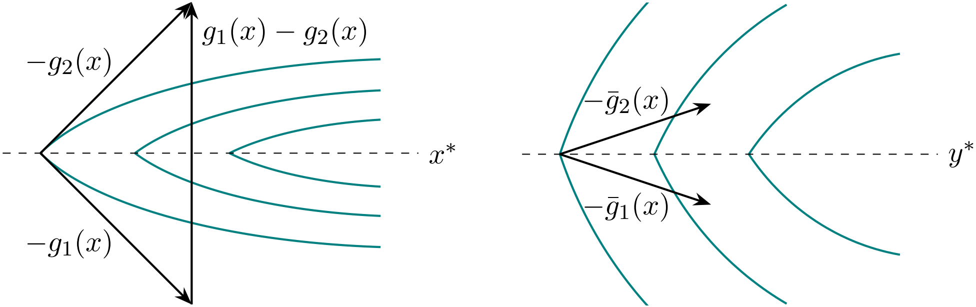
Problem statement

Let us consider the following constrained optimization problem:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) \quad \text{subject to} \quad \langle \boldsymbol{a}_i, \boldsymbol{x} \rangle = b_i, \quad i = 1, \dots, m, \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function, $\boldsymbol{a}_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}$, $i = 1, \dots, m$.

How can we solve this problem with the $r(\alpha)$-algorithm?



(a) before the space dilation (b) after the space dilation

Figure: Space dilation in the direction of the vector $g_1(x) - g_2(x)$ allows for the subgradients in the transformed space to form an acute angle.

Definition of the $r(\alpha)$-algorithm

Let $\boldsymbol{x}_0 \in \mathbb{R}^n$, $h_0 > 0$, $\boldsymbol{B}_0 := I \in \mathbb{R}^{n\times n}$, and $\alpha > 1$ be given. Then, the $r(\alpha)$-algorithm is an iterative procedure that constructs sequences $(\boldsymbol{x}_k)_{k\in\mathbb{N}} \subset \mathbb{R}^n$, $(\boldsymbol{h}_k)_{k\in\mathbb{N}} \subset (0,\infty)$, $(\boldsymbol{\xi}_k)_{k\in\mathbb{N}} \subset \mathbb{R}^n$, $(\boldsymbol{\eta}_k)_{k\in\mathbb{N}} \subset \mathbb{R}^n$, and $(\boldsymbol{B}_k)_{k\in\mathbb{N}} \subset \mathbb{R}^{n\times n}$ according to the following scheme:

$$\boldsymbol{x}_{k+1} := \boldsymbol{x}_k - \boldsymbol{h}_k \boldsymbol{B}_k \boldsymbol{\xi}_k, \quad \boldsymbol{B}_{k+1} := \boldsymbol{B}_k \boldsymbol{R}_\beta(\boldsymbol{\eta}_k), \quad k = 0, 1, 2, \dots, \tag{2}$$

where

$$\boldsymbol{\xi}_k := \frac{\boldsymbol{B}_k^\top g_f(\boldsymbol{x}_k)}{\left\|\boldsymbol{B}_k^\top g_f(\boldsymbol{x}_k)\right\|}, \quad \boldsymbol{h}_k \geq \boldsymbol{h}_k^* := \arg\min_{h\geq 0} f(\boldsymbol{x}_k - h\boldsymbol{B}_k\boldsymbol{\xi}_k), \tag{3}$$

$$\boldsymbol{\eta}_k := \frac{\boldsymbol{B}_k^\top \boldsymbol{r}_k}{\left\|\boldsymbol{B}_k^\top \boldsymbol{r}_k\right\|}, \quad \boldsymbol{r}_k := g_f(\boldsymbol{x}_{k+1}) - g_f(\boldsymbol{x}_k), \tag{4}$$

$$\boldsymbol{R}_\beta(\boldsymbol{\eta}_k) := \boldsymbol{I} + (\beta - 1)\boldsymbol{\eta}_k \boldsymbol{\eta}_k^\top, \quad \beta := \frac{1}{\alpha}, \tag{5}$$

provided that no division by zero occurs.

Common approach: penalty functions

The most common approach to transforming constrained problems into unconstrained ones is to use penalty functions. For example, one can use the following penalty functions:

1. Quadratic penalty:

$$f_2(\boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{i=1}^{m} Q_i(\langle \boldsymbol{a}_i, \boldsymbol{x} \rangle - b_i)^2, \qquad (6)$$

   where $Q_i > 0$, $i = 1, \ldots, m$ are the penalty parameters.

2. Absolute value penalty:

$$f_1(\boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{i=1}^{m} P_i|\langle \boldsymbol{a}_i, \boldsymbol{x} \rangle - b_i|, \qquad (7)$$

   where $P_i > 0$, $i = 1, \ldots, m$ are the penalty parameters.

Common approach: penalty functions

Drawback: Having to choose the penalty parameters P_i and Q_i can be challenging.

For example:

1. Quadratic penalty: the resulting problem is not equivalent to the original one for any finite values of the penalty parameters Q_i. In practice, one can use large values of Q_i to achieve a good approximation of the original problem.
2. Absolute value penalty: there exist finite values of the penalty parameters P_i that guarantee the equivalence of the original problem and the penalized one, but they are unknown in advance (can be determined from the optimal Lagrange multipliers).

Projections method

Instead, we can use the $r(\alpha)$-algorithm with projections to solve the constrained problem.

1. Choose some initial point $\boldsymbol{x}_0 \in \mathbb{R}^n$ and set $\boldsymbol{B}_0 = \boldsymbol{I} \in \mathbb{R}^{n \times n}$.
2. Execute the projections method for $i = 1, \ldots, m$ to find a point $\boldsymbol{x}_m$ that satisfies the constraints $\langle \boldsymbol{a}_i, \boldsymbol{x}_m \rangle = b_i$, $i = 1, \ldots, m$ and the corresponding matrix $\boldsymbol{B}_m$.
3. Use the $r(\alpha)$-algorithm to minimize the function f without constraints, using the point $\boldsymbol{x}_m$ and the matrix $\boldsymbol{B}_m$ as the starting parameters.

Projections method

1. At the iteration $i = 1, \ldots, m$ compute the direction of the space dilation $\boldsymbol{\xi}_i$ by the formula:

$$\boldsymbol{\xi}_i = \frac{\boldsymbol{B}_{i-1}^{\mathsf{T}} \boldsymbol{a}_i}{\left\| \boldsymbol{B}_{i-1}^{\mathsf{T}} \boldsymbol{a}_i \right\|}. \tag{8}$$

2. Update the vector $\boldsymbol{x}$ and the matrix $\boldsymbol{B}$ using the formulas:

$$\boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \frac{b_i - \langle \boldsymbol{a}_i, \boldsymbol{x}_{i-1} \rangle}{\left\| \boldsymbol{B}_{i-1}^{\mathsf{T}} \boldsymbol{a}_i \right\|} \boldsymbol{\xi}_i, \tag{9}$$

$$\boldsymbol{B}_i = \boldsymbol{B}_{i-1} \left(\boldsymbol{I} - \boldsymbol{\xi}_i \boldsymbol{\xi}_i^{\mathsf{T}} \right). \tag{10}$$

3. If $i < m$, then increment i and go to step 1, otherwise stop the algorithm.

Applications: Basis Pursuit

Basis pursuit is a well-known optimization problem used to find the sparsest solution to an underdetermined system of linear equations. It is formulated as follows:

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},$$

where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, $n > m$.

$$a_{ij} \sim \mathcal{N}(0, 1), \quad i = 1, \ldots, m, \quad j = 1, \ldots, n.$$

Applications: Basis Pursuit

Table: Comparison of penalized and projections methods for $n = 1000$.

			absolute value penalty			projections method		
σ	m	test	$\|\hat{\boldsymbol{x}} - \boldsymbol{x}^*\|_2$	t, s	itn	$\|\hat{\boldsymbol{x}} - \boldsymbol{x}^*\|_2$	t, s	itn
0.01	280	1	1.04e-06	33.6	10673	1.02e-06	16.9	7279
		2	1.11e-06	33.4	10657	1.10e-06	16.9	7275
		3	1.07e-06	33.4	10661	9.93e-07	17.1	7289
0.05	420	1	1.37e-06	35.0	10980	1.42e-06	13.9	5860
		2	1.25e-06	34.7	10942	1.20e-06	13.9	5862
		3	1.19e-06	35.3	10982	1.17e-06	13.9	5865
0.1	550	1	1.12e-06	37.9	11163	1.04e-06	11.3	4569
		2	1.19e-06	37.7	11161	1.19e-06	11.2	4546
		3	1.30e-06	37.1	11173	1.12e-06	11.2	4544

The MAXQUAD problem is a well-known minimax test problem in the field of optimization, that was proposed by C. Lemaréchal. Consider the following optimization problem for $\boldsymbol{x} \in \mathbb{R}^{10}$:

$$\min_{\boldsymbol{x} \in \mathbb{R}^{10}} f(\boldsymbol{x}) = \min_{\boldsymbol{x} \in \mathbb{R}^{10}} \max_{j=1,\dots,5} f_j(\boldsymbol{x}), \tag{11}$$

where $f_j(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{Q}^{(j)} \boldsymbol{x} \rangle - \langle \boldsymbol{x}, \boldsymbol{c}^{(j)} \rangle$ and matrices $\boldsymbol{Q}^{(j)} \in \mathbb{R}^{10 \times 10}$ and vectors $\boldsymbol{c}^{(j)} \in \mathbb{R}^{10}$ are defined as follows:

$$c_i^{(j)} = e^{i/j} \sin ij,$$

$$Q_{ik}^{(j)} = \begin{cases} e^{i/k} \cos ik \sin j & \text{for } k = i+1, \dots, 10, \\ Q_{ki}^{(j)} & \text{for } k = 1, \dots, i-1, \\ |\sin j| i/10 + \sum_{l \neq i} |Q_{il}^{(j)}| & \text{for } k = i. \end{cases}$$

Applications: MAXQUAD problem

For the experiments we generated a random matrix $\boldsymbol{A}$ with 9 rows and full row rank and set $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}^*$, where $\boldsymbol{x}^*$ is the numerical solution of the original unconstrained problem as seen in [Rzhevskyi, 1993]:

$$\boldsymbol{A} = \begin{bmatrix} 3 & 6 & -9 & -6 & -6 & -2 & 0 & -5 & -3 & 3 \\ -8 & -3 & -2 & 5 & 8 & -4 & 4 & -1 & 0 & 7 \\ -4 & 6 & 6 & -9 & -6 & 8 & 5 & -2 & -9 & -8 \\ -3 & -1 & 8 & 6 & -5 & 0 & 1 & -8 & -8 & -2 \\ 0 & -9 & 1 & -6 & 2 & -7 & -9 & -9 & -5 & -4 \\ 0 & -6 & -3 & 2 & 5 & -9 & 5 & -6 & 3 & 1 \\ 2 & -5 & -3 & -5 & 6 & -6 & 3 & -5 & -1 & 5 \\ 6 & -6 & 6 & 4 & 7 & 8 & -4 & 0 & -6 & -9 \\ -4 & -9 & 8 & -5 & -7 & 7 & -6 & -7 & 1 & 4 \end{bmatrix}, \quad \boldsymbol{x}^* = \begin{bmatrix} -0.1262570 \\ -0.0343781 \\ -0.0068572 \\ 0.0263607 \\ 0.0672950 \\ -0.2783990 \\ 0.0742187 \\ 0.1385240 \\ 0.0840313 \\ 0.0385803 \end{bmatrix}.$$

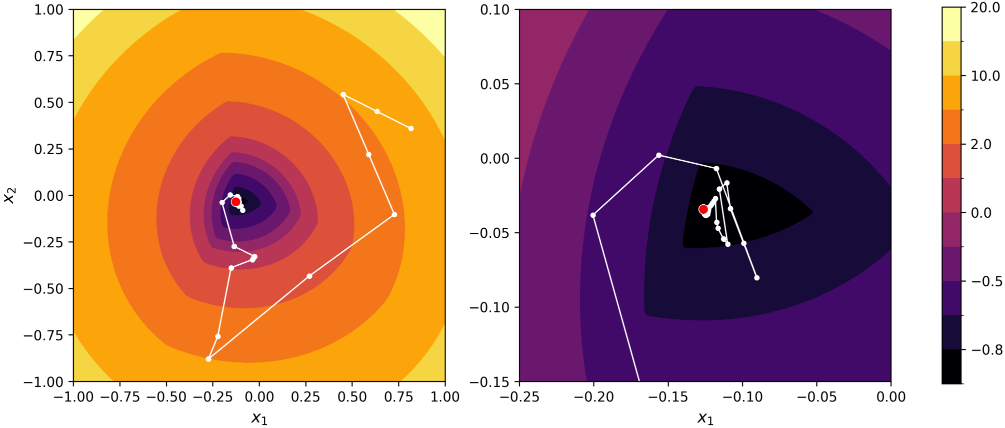


Figure: Plots of the function f and the record sequence of the $r(\alpha)$-algorithm in the plane of the first two components of the solution for the MAXQUAD problem with $m = 4$; the red dot indicates the solution $\boldsymbol{x}^*$.

Applications: MAXQUAD problem

Table: Performance of the $r(\alpha)$-algorithm with projections for different numbers of constraints m and values of α.

	$\alpha = 2$			$\alpha = 4$		
m	$f(\hat{\boldsymbol{x}})$	itn	nfg	$f(\hat{\boldsymbol{x}})$	itn	nfg
1	-0.8414083	252	279	-0.8414083	160	288
2	-0.8414083	235	269	-0.8414083	128	203
3	-0.8414083	198	245	-0.8414083	125	221
4	-0.8414083	187	222	-0.8414083	114	196
5	-0.8414083	160	196	-0.8414083	87	154
6	-0.8414083	126	161	-0.8414083	71	153
7	-0.8414083	91	113	-0.8414083	54	115
8	-0.8414069	60	79	-0.8414069	31	71
9	-0.8414056	31	45	-0.8414056	16	41

Conclusion

- We discussed a modification of the $r(\alpha)$-algorithm for solving convex optimization problems with linear equality constraints and tested it on the basis pursuit and MAXQUAD problems.
- Overall, the proposed modification of the $r(\alpha)$-algorithm provides an effective approach for solving convex optimization problems with linear equality constraints, and it can be applied to a wide range of problems in various fields.
- In future work, we plan to explore the application of the $r(\alpha)$-algorithm to other classes of constrained convex optimization problems and provide a more detailed theoretical analysis of the proposed projection method.

Thank you for your attention!

References

Amelunxen, D., M. Lotz, M. B. McCoy, and J. A. Tropp. "Living on the Edge: Phase Transitions in Convex Programs with Random Data". In: *Information and Inference: A Journal of the IMA* 3.3 (2014), pp. 224–294.

Bonnans, J. F., J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal. *Numerical Optimization*. Springer Berlin Heidelberg, 2006.

Candes, E. J. and T. Tao. "Decoding by Linear Programming". In: *IEEE Transactions on Information Theory* 51.12 (2005), pp. 4203–4215.

Rockafellar, R. T. *Convex Analysis*. Princeton: Princeton University Press, 1970.

Rzhevskyi, S. V. *Monotonnye metody vypuklogo programmirovanija [Monotone Methods for Convex Programming]*. In Russian. Kyiv: Naukova Dumka, 1993.

Shor, N. Z. *Minimization Methods for Non-Differentiable Functions*. Berlin: Springer, 1985.

Stetsyuk, P. "K metodam ellipsoidov [On the Ellipsoid Methods]". In: *Teoriia optymalnykh rishen* (1999). In Russian, pp. 27–33.

Stetsyuk, P., A. Fischer, and O. Pichugina. "A Penalty Approach to Linear Programs with Many Two-Sided Constraints". In: *Lecture Notes in Computer Science*. Vol. 12755. 2021, pp. 206–217.