Linear Programming Solvers and r-Algorithm for Sparse Signal Reconstruction

Anastasiia Storozhenko [1,2] Petro Stetsyuk [1]

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

[2]Kyiv Academic University, Kyiv, Ukraine

ITTA 2026

What is compressed sensing?

$$\boldsymbol{Ax} = \boldsymbol{b}$$

$\boldsymbol{A} \in \mathcal{M}_{m\times n}(\mathbb{R})$ — measurement matrix,

$\boldsymbol{x} \in \mathbb{R}^n$ — unknown sparse vector,

$\boldsymbol{b} \in \mathbb{R}^m$ — measurement vector.

$$\boldsymbol{x} = \,?$$

$$\boxed{m < n}$$

l_1 minimization

$$\text{minimize} \quad \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}. \tag{P_1}$$

Is P_1 equivalent to the original problem?

Recovery guarantees for l_1 minimization

Various conditions exist that guarantee the equivalence of the l_1 minimization problem and the original sparse recovery problem, but in this work we use the following probabilistic result for Gaussian measurement matrices[1]:

Theorem

Let $\boldsymbol{x}^* \in \mathbb{R}^n$ be a fixed vector and $p \in (0,1)$. Suppose $\boldsymbol{A} \in M_{m\times n}(\mathbb{R})$ is a matrix with independent $\mathcal{N}(0,1)$-distributed entries and $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}^*$. Then

$$m \le \delta(\mathcal{D}(\|\cdot\|_1, \boldsymbol{x}^*)) - a_\eta\sqrt{n} \implies \mathbb{P}\{\boldsymbol{x}^* \text{ is the unique solution of } (P_1)\} \le \eta$$
$$m \ge \delta(\mathcal{D}(\|\cdot\|_1, \boldsymbol{x}^*)) + a_\eta\sqrt{n} \implies \mathbb{P}\{\boldsymbol{x}^* \text{ is the unique solution of } (P_1)\} \ge 1-\eta,$$

where $a_\eta = \sqrt{8\log(4/\eta)}$.

[1] Amelunxen et al., "Living on the Edge" (Sept. 2014).

Linear programming approach for l_1 minimization

Problem (P_1) can be reformulated as the following linear programming (LP) problem:

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{n} y_i && \text{(LP1)}\\
\text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, && \text{(LP2)}\\
& -y_i \leq x_i \leq y_i, && \text{(LP3)}\\
& y_i \geq 0, i = 1, \ldots, n. && \text{(LP4)}
\end{aligned}$$

LP solvers for basis pursuit

We tested the following optimization solvers on the problem (LP1)–(LP4):

- CPLEX, a commercial solver developed by IBM;
- HiGHS, an open-source solver;
- Gurobi, a commercial solver developed by Gurobi Optimization, LLC;
- Clarabel, a newer open-source solver.

All experiments were run in Python, with the first three solvers accessed through the AMPL API, and Clarabel accessed through the cvxpy library.

Table: LP solvers comparison for $n = 5000$.

σ	m	test	CPLEX		HiGHS		Gurobi		Clarabel	
			t, s	$\|\hat{x} - x^*\|_2$	t, s	$\|\hat{x} - x^*\|_2$	t, s	$\|\hat{x} - x^*\|_2$	t, s	$\|\hat{x} - x^*\|_2$
0.01	790	0	31.6	1.73e-12	48	1.02e-08	22.4	1.22e-12	11.5	8.16e-08
		1	29.6	4.85e-12	44.4	5.61e-12	5.7	5.8e-12	11.6	9.65e-08
		2	31	3.94e-12	46.9	5.27e-10	5.76	6.9e-12	12.8	2.95e-07
		3	31.2	5.9e-12	44.1	3.98e-07	5.8	4.06e-12	12.5	1.13e-08
		4	29.3	4.22e-12	46.9	8.48e-08	5.81	6.53e-12	11.3	9e-08
0.05	1510	0	84.7	1.11e-11	2183	1.22e-10	81.1	6.69e-12	38.8	5.49e-08
		1	84.9	1.17e-11	169	2.73e-09	80.5	2.24e-12	39.8	5.76e-08
		2	84.8	1.17e-11	192	2.89e-10	29.5	1.16e-11	37.8	1.53e-07
		3	84	1.19e-11	170	8.73e-09	80.6	3.16e-12	38.6	1.33e-06
		4	84.7	1.04e-11	163	5.73e-08	80.5	3.49e-11	41.1	6.12e-08
0.1	2140	0	158	1.6e-11	589	1.58e-09	143	1.3e-11	89.8	8.53e-07
		1	159	1.53e-11	396	9.44e-11	151	8.23e-12	90.3	1.08e-07
		2	164	1.43e-11	383	6.54e-09	50.5	1.84e-11	89.8	3.64e-07
		3	160	1.62e-11	590	7.66e-08	139	8.75e-12	89.8	1.37e-07
		4	159	1.54e-11	384	3.19e-10	140	1.45e-11	89.6	4.75e-06

$r(\alpha)$-algorithm for basis pursuit

In this work, we also apply the $r(\alpha)$-algorithm, which is an optimization method for unconstrained convex problems developed by N.Z. Shor[2], to solve the basis pursuit problem (P_1). We can do so by considering the following penalized objective functions:

$$f_1(\boldsymbol{x}) = \sum_{i=1}^{n} |x_i| + \sum_{i=1}^{m} P_i \left| \sum_{j=1}^{n} a_{ij} x_j - b_i \right|, \tag{1}$$

$$f_2(\boldsymbol{x}) = \sum_{i=1}^{n} |x_i| + \sum_{i=1}^{m} Q_i \left(\sum_{j=1}^{n} a_{ij} x_j - b_i \right)^2. \tag{2}$$

[2] Shor, *Nondifferentiable Optimization and Polynomial Problems* (1998).

Choosing the penalty parameters

It can be shown that for the linear penalty function f_1, there exist finite values of the penalty parameters P_i that guarantee the equivalence of the original problem and the penalized one:

Proposition

Let λ_i^*, $i = 1, \ldots, m$, be the optimal Lagrange multipliers for the constraints in the basis pursuit problem. If the penalty parameters P_i in (1) satisfy $P_i \geq |\lambda_i^*|$ for all $i = 1, \ldots, m$, then $X^* \subset X_1^*$. Moreover, if $P_i > |\lambda_i^*|$ for all $i = 1, \ldots, m$, then $X^* = X_1^*$.

At the same time, for the quadratic penalty function f_2, the resulting problem is not equivalent to the original one for any finite values of the penalty parameters Q_i.

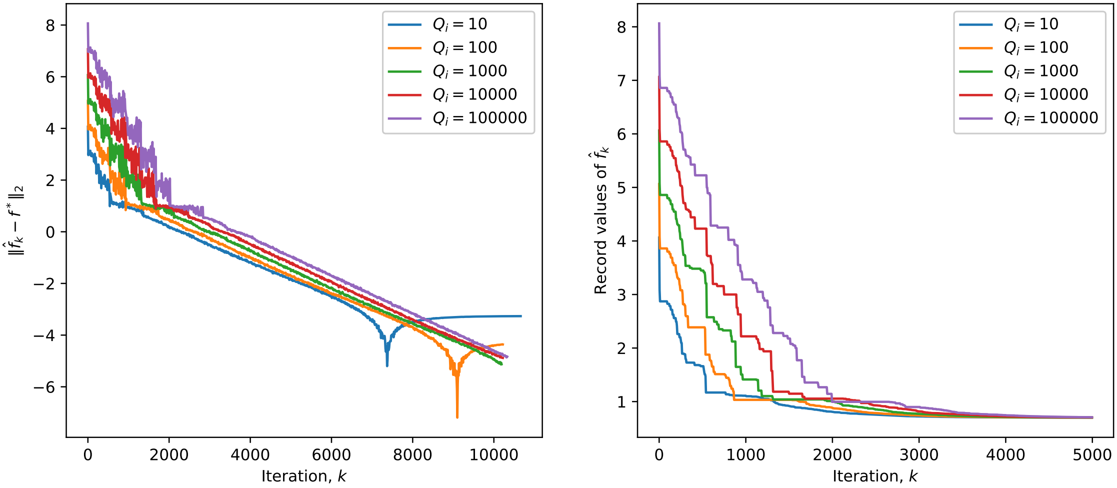

Figure: Convergence of the $r(\alpha)$-algorithm with quadratic penalty for different values of the penalty parameter Q. The plot on the left shows the deviation of the objective value from the optimal value per iteration, and the plot on the right shows the record values of the objective function.

Table: Quadratic penalty results for $n = 1000$.

			Clarabel			ralgb5a		
s	m	test	$\lvert\hat{f} - f^*\rvert$	$\lVert\hat{\boldsymbol{x}} - \boldsymbol{x}^*\rVert_2$	t, s	$\lvert\hat{f} - f^*\rvert$	$\lVert\hat{\boldsymbol{x}} - \boldsymbol{x}^*\rVert_2$	t, s
0.01	280	0	9.78e-07	6.55e-07	0.424	5.62e-05	3.03e-06	8.37
		1	5.97e-07	6.28e-07	0.363	4.15e-05	2.67e-06	8.51
		2	9.63e-07	6.33e-07	0.418	3.91e-05	2.71e-06	8.65
		3	6.47e-07	5.25e-07	0.368	3.98e-05	2.11e-06	8.02
		4	4.8e-07	6.27e-07	0.364	3.32e-05	2.1e-06	8.38
0.05	420	0	4.29e-06	1.36e-06	0.804	7.87e-05	6.08e-06	13.1
		1	2.82e-06	1.12e-06	0.695	0.000121	6.11e-06	12.4
		2	2.82e-06	1.22e-06	0.696	0.000103	6.78e-06	12.2
		3	2.74e-06	1e-06	0.68	6.83e-05	4.38e-06	12.1
		4	2.92e-06	1.06e-06	0.682	8.17e-05	5.25e-06	12.5
0.1	550	0	4.45e-06	1.45e-06	1.09	0.000103	6.88e-06	16.8
		1	6.46e-06	1.63e-06	1.24	9.3e-05	8.02e-06	17.5
		2	4.62e-06	1.66e-06	1.08	0.000104	6.84e-06	16.8
		3	5.29e-06	1.46e-06	1.08	0.000112	7.11e-06	16.6
		4	3.99e-06	1.65e-06	1.09	8.65e-05	6.89e-06	16.5

Conclusion

- We compared several LP solvers for the reformulated basis pursuit problem and found that with default parameters, Gurobi and CPLEX provide the best balance of accuracy and speed, while Clarabel is the fastest but with lower accuracy.
- We applied the $r(\alpha)$-algorithm to solve the basis pursuit problem with different penalty functions and discovered that in practice it is best to use the quadratic penalty function with sufficiently large penalty parameters, which results in the best convergence speed for this algorithm.
- In future work, we plan to further investigate the potential of the $r(\alpha)$-algorithm for solving the basis pursuit problem through algorithmic modifications and parameter tuning.

References I

- Amelunxen, Dennis, Martin Lotz, Michael B. McCoy, and Joel 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 (Sept. 2014), pp. 224–294. ISSN: 2049-8764. DOI: 10.1093/imaiai/iau005. (Visited on 03/10/2026).
- Boyd, Stephen P. and Lieven Vandenberghe. *Convex Optimization*. Cambridge, UK ; New York: Cambridge University Press, 2004. ISBN: 978-0-521-83378-3.
- Candes, E.J. and T. Tao. "Decoding by linear programming". In: *IEEE Transactions on Information Theory* 51.12 (2005), pp. 4203–4215. DOI: 10.1109/TIT.2005.858979.
- Candes, Emmanuel J. and Yaniv Plan. *A Probabilistic and RIPless Theory of Compressed Sensing*. Nov. 2010. DOI: 10.48550/arXiv.1011.3854. arXiv: 1011.3854 [cs]. (Visited on 03/06/2026).
- Chandrasekaran, Venkat, Benjamin Recht, Pablo A. Parrilo, and Alan S. Willsky. "The Convex Geometry of Linear Inverse Problems". In: *Foundations of Computational Mathematics* 12.6 (Dec. 2012), pp. 805–849. ISSN: 1615-3375, 1615-3383. DOI: 10.1007/s10208-012-9135-7. arXiv: 1012.0621 [math]. (Visited on 02/20/2026).
- Donoho, D. "Compressed Sensing". In: *Information Theory, IEEE Transactions on* 52 (May 2006), pp. 1289–1306. DOI: 10.1109/TIT.2006.871582.
- Donoho, David L. and Jared Tanner. *Counting Faces of Randomly-Projected Polytopes When the Projection Radically Lowers Dimension*. Sept. 2006. DOI: 10.48550/arXiv.math/0607364. arXiv: math/0607364. (Visited on 03/10/2026).

References II

- Foucart, Simon and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. Applied and Numerical Harmonic Analysis. New York, NY: Springer, 2013. ISBN: 978-0-8176-4947-0 978-0-8176-4948-7. DOI: 10.1007/978-0-8176-4948-7. (Visited on 02/21/2026).
- Goulart, Paul J. and Yuwen Chen. *Clarabel: An Interior-Point Solver for Conic Programs with Quadratic Objectives*. May 2024. DOI: 10.48550/arXiv.2405.12762. arXiv: 2405.12762 [math]. (Visited on 02/20/2026).
- Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. Accessed: 2026-03-06. URL: https://www.gurobi.com.
- Huangfu, Q. and J. A. J. Hall. “Parallelizing the dual revised simplex method”. In: *Mathematical Programming Computation* 10.1 (2018), pp. 119–142. DOI: 10.1007/s12532-017-0130-5.
- IBM ILOG. *CPLEX Optimization Studio Documentation*. Accessed: 2026-03-06. URL: https://www.ibm.com/docs/en/icos.
- Shor, Naum Z. *Nondifferentiable Optimization and Polynomial Problems*. Ed. by Panos Pardalos and Reiner Horst. Vol. 24. Nonconvex Optimization and Its Applications. Boston, MA: Springer US, 1998. ISBN: 978-1-4419-4792-5 978-1-4757-6015-6. DOI: 10.1007/978-1-4757-6015-6. (Visited on 03/01/2026).
- Stetsyuk, Petro, Andreas Fischer, and Oksana Pichugina. “A Penalty Approach to Linear Programs with Many Two-Sided Constraints”. In: *Mathematical Optimization Theory and Operations Research*. Ed. by Panos Pardalos, Michael Khachay, and Alexander Kazakov. Cham: Springer International Publishing, 2021, pp. 206–217. ISBN: 978-3-030-77876-7. DOI: 10.1007/978-3-030-77876-7_14.

Definition

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:

$$\boldsymbol{x}_{k+1} := \boldsymbol{x}_k - \boldsymbol{h}_k \boldsymbol{B}_k \boldsymbol{\xi}_k, \quad \boldsymbol{B}_{k+1} := \boldsymbol{B}_k \boldsymbol{R}_\beta(\boldsymbol{\eta}_k), \quad k = 0, 1, 2, \dots, \tag{3}$$

where

$$\boldsymbol{\xi}_k := \frac{\boldsymbol{B}_k^\top g_f(\boldsymbol{x}_k)}{\left\|\boldsymbol{B}_k^\top g_f(\boldsymbol{x}_k)\right\|}, \quad \boldsymbol{h}_k \geq \boldsymbol{h}_k^* := \arg\min_{h\geq 0} f(\boldsymbol{x}_k - h\boldsymbol{B}_k\boldsymbol{\xi}_k), \tag{4}$$

$$\boldsymbol{\eta}_k := \frac{\boldsymbol{B}_k^\top \boldsymbol{r}_k}{\left\|\boldsymbol{B}_k^\top \boldsymbol{r}_k\right\|}, \quad \boldsymbol{r}_k := g_f(\boldsymbol{x}_{k+1}) - g_f(\boldsymbol{x}_k), \tag{5}$$

$$\boldsymbol{R}_\beta(\boldsymbol{\eta}_k) := \boldsymbol{I} + (\beta - 1)\boldsymbol{\eta}_k \boldsymbol{\eta}_k^\top, \quad \beta := \frac{1}{\alpha}, \tag{6}$$

provided that no division by zero occurs.