# Supplementary Material: Foundations of Correlated Mutations for Integer Programming

Ofer M. Shir
Tel-Hai College and Migal Institute
Upper Galilee, Israel
ofersh@telhai.ac.il

Michael Emmerich
University of Jyväskylä
Finland
michael.t.m.emmerich@jyu.fi

The source code supporting this research is publicly available at https://github.com/ofersh/correlated-mutations-for-integer-programming .

The following provides supplementary technical details that were omitted from the main text due to space constraints..

## A MULTIMODALITY OF CONVEX IQP

Quadratic forms arise naturally in optimization and are central to problems involving convex and nonconvex objectives. A general quadratic form in two variables is given by

$$f(x,y) = \mathbf{x}^\top A\mathbf{x} = ax^2 + 2bxy + cy^2,$$

where $\mathbf{x} = [x\ y]^\top$ and $A = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$ is a symmetric matrix. If $A$ is positive definite, the function is convex and has a unique global minimum. In the continuous domain, this ensures unimodality—any local minimum is also global, and the level sets are smooth ellipses aligned with the eigenvectors of $A$.

However, this well-behaved structure does not necessarily extend to integer domains. When optimization is restricted to integer points, even convex quadratic functions can exhibit multimodal behavior. Discretization introduces alignment mismatches between the smooth continuous minimum and the grid structure, especially when the optimum lies between integer coordinates or the function exhibits high anisotropy.

To illustrate this, we consider the quadratic form $f(x,y) = \mathbf{x}^\top A\mathbf{x}$ with a symmetric positive definite matrix of high condition number:

$$A = \begin{bmatrix} 100 & 1 \\ 1 & 1 \end{bmatrix}.$$

We rotate the matrix by 30° using a similarity transformation $A' = R^\top AR$, where

$$R = \begin{bmatrix} \cos(30^\circ) & -\sin(30^\circ) \\ \sin(30^\circ) & \cos(30^\circ) \end{bmatrix}.$$

The resulting rotated matrix is

$$A' = \begin{bmatrix} 8.62 & -3.40 \\ -3.40 & 2.38 \end{bmatrix}.$$

The explicit quadratic function plotted is then:

$$f(x,y) = 8.62\,x^2 - 6.80\,xy + 2.38\,y^2,$$


*FOGA '25, August 27–29, 2025, Leiden, Netherlands*

ACM ISBN 979-8-4007-1859-5/2025/08
https://doi.org/10.1145/3729878.3746698

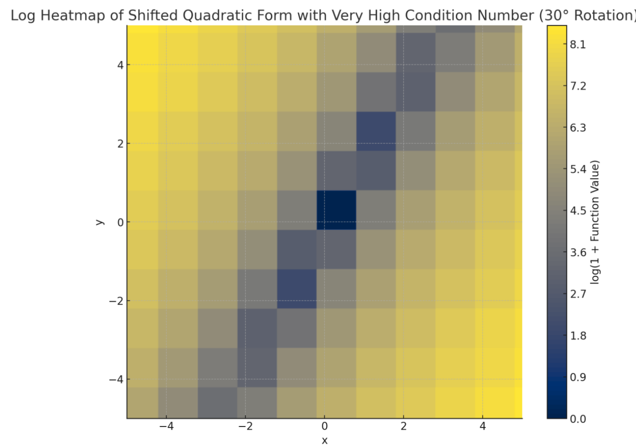


**Figure 1: Log-scaled heatmap of a shifted, rotated quadratic form with a high condition number evaluated over an integer lattice. Multimodal behavior arises from discretization effects.**

evaluated over integer grid points after a small shift of the minimum to (0.3, 0.3) to break alignment with the grid.

$$f(x,y) = 8.62\,x^2 - 6.80\,xy + 2.38\,y^2,$$

evaluated over integer grid points after a small shift of the minimum to (0.3, 0.3) to break alignment with the grid. We validate that it is positive definite and elaborate on our calculation process in the Appendix. As shown in Figure 1, the resulting landscape—visualized in logarithmic scale—exhibits distinct local minima, clearly demonstrating how multimodality can emerge in integer quadratic optimization problems.

Let us double check that the problem is convex (and thus unimodal) in the continuous case : The Hessian matrix $H$ of $f$ is

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} = \begin{bmatrix} 17.24 & -6.80 \\ -6.80 & 4.76 \end{bmatrix},$$

with eigenvalues

$$\lambda_1 \approx 20.23, \quad \lambda_2 \approx 1.77 .$$

As both eigenvalues are positive, $H$ is **positive definite**.

The plot in Figure 1 clearly shows that the problem becomes multimodal (i.e., it has several local optima in von Neumann and in the Moore neighborhood). It is easy to prove that this cannot occur in the uncorrelated quadratic case; the observed effect is due to the rotation.

## B CORRELATION VIA LASSO COEFFICIENTS

Since one of our challenges was to consider appropriate measures that respect the $\ell_1$-norm, we also utilize the Least Absolute Shrinkage and Selection Operator (LASSO) [3] to measure the variables' correlations over the integer lattice. This measure is known for its ability to quantify the strength of the relationship between predictor-to-target variables in the context of $\ell_1$ regularization. Figure 2 presents the LASSO correlation measure for the rotated 2D populations that were considered in the analysis of Figure 7 in the main paper.

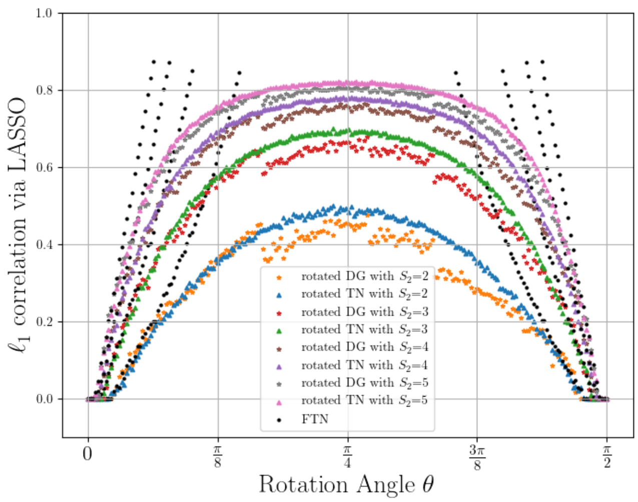


**Figure 2: Additional correlation computation via the LASSO coefficients of the 2D populations examined in Figure 7 in the main paper.**

## C IES DESCRIPTION

This ES was originally defined to treat altogether real-valued, integer ordinal, and categorical decision variables [2]. We discard herein the treatment of continuous and categorical variables. Like most ESs, its operation is well defined by its self-adaptive mutation operator (`corrMutate`). The recombination operator of the IES is the discrete recombination, which randomly assigns pairs as 'parents' and then randomly chooses a component at a vector position of the child individual from one of the two parents at the same position. Selection is non-elitist ranked-based deterministic with a population of $(15, 100)$. It is summarized as Algorithm 1.

**ies**$(\mu, \lambda,$ type)
$t \leftarrow 0$
$P(t) \leftarrow$ randIntUniform$(\mu)$ /* forming $\mu$ individuals, each with decision variables + strategy parameters*/
evaluate$(P(t))$
**repeat**
  $P'(t) \leftarrow$ recombine$(P(t))$ /* forming $\lambda$ offspring by repeatedly drawing $\frac{\lambda}{2}$ pairs of parents at random */
  $P''(t) \leftarrow$ mutate$(P'(t),$ type) /* calling corrMutate, which also self-adapts the strategy parameters */
  evaluate$(P''(t))$
  $P(t+1) \leftarrow$ select$(P''(t))$ /* deterministically selecting the top $\mu$ individuals post-sorting */
  $t \leftarrow t + 1$
**until** evaluation budget is exhausted
**return** { best individual found }

**Algorithm 1:** $(\mu, \lambda)$ Integer Evolution Strategy

The generalized mutation operator for a standard IES is presented as Algorithm 3, generically prescribed for the usage of either the DG or TN distributions, subject to rotations (correlations).

## REFERENCES

[1] Thomas Bäck. 1996. *Evolutionary Algorithms in Theory and Practice.* Oxford University Press, New York, NY, USA.
[2] Thomas Bäck and Martin Schütz. 1995. Evolution Strategies for Mixed Integer Optimization of Optical Multilayer Systems. In *Evolutionary Programming IV – Proc. Fourth Annual Conf. Evolutionary Programming*. The MIT Press, 33–51.
[3] Robert Tibshirani. 1996. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 58, 1 (1996), 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

**ies::genUncorrelatedMutation**($\vec{S}$, type)

$n \longleftarrow$ len$\left(\vec{S}\right)$, $\quad \vec{z} := \vec{0} \in \mathbb{R}^n$

**if** type==DG **then**

  **for** $i = 1, \ldots, n$ **do**

    $p_i \longleftarrow 1 - \frac{S_i/n}{\sqrt{\left(1+(S_i/n)^2\right)}+1}$

    $z_i \longleftarrow \mathcal{G}(0, p_i)$

  **end**

**else**

  /* default TN */

  **for** $i = 1, \ldots, n$ **do**

    $\sigma_i \longleftarrow \sqrt{\frac{\pi}{2}} \cdot \frac{S_i}{n}$

    $z_i \longleftarrow \sigma_i \cdot \mathcal{N}(0, 1)$

  **end**

**end**

**return** $\{\vec{z}\}$

**Algorithm 2:** Procedure to generate uncorrelated $n$-dimensional mutations, either DG- or TN-based (determined by type), using geometric ($\mathcal{G}$) or Normal ($\mathcal{N}$) generators. While the step-sizes in $\vec{\sigma}$ represent standard deviations in the Normal distribution, they represent the expected $\ell_1$-norm for the DG-based vectors – not directly comparable.

**ies::corrMutate**($\vec{x}$, $\vec{S}$, $\vec{\alpha}$, $n$, type)

$\mathcal{N}_g \leftarrow \mathcal{N}(0, 1)$, $\tau_g \leftarrow \frac{1}{\sqrt{2 \cdot n}}$, $\tau_\ell \leftarrow \frac{1}{\sqrt{2 \cdot \sqrt{n}}}$

**for** $i = 1, \ldots, n$ **do**

  $S'_i \longleftarrow S_i \cdot \exp\left\{\tau_g \cdot \mathcal{N}_g + \tau_\ell \cdot \mathcal{N}_i(0, 1)\right\}$

**end**

**for** $j = 1, \ldots, n \cdot (n-1)/2$ **do**

  $\alpha'_j \longleftarrow \alpha_j + \beta \cdot \mathcal{N}_j(0, 1)$

**end**

$\vec{z}_u \longleftarrow$ genUncorrelatedMutation$\left(\vec{S}', \text{type}\right)$

$\vec{z} \longleftarrow$ round (rotate ($\vec{z}_u$, $\vec{\alpha}'$))

**if** type==DG **then**

  $\vec{z}_g \longleftarrow$ genUncorrelatedMutation$\left(\vec{S}', \text{type}\right)$

  $\vec{z}'_g \longleftarrow$ round (rotate ($\vec{z}_g$, $\vec{\alpha}'$))

  $\vec{z} \longleftarrow \vec{z} - \vec{z}'_g$ /* difference of two geometric samples */

**end**

$\vec{x}' \longleftarrow \vec{x} + \vec{z}$

**return** $\left\{\vec{x}', \vec{S}', \vec{\alpha}'\right\}$

**Algorithm 3:** The self-adaptive mutation operator utilized by the ies: $\{\vec{\sigma}, \vec{\alpha}\}$ are the strategy parameters: step-sizes and rotation angles, respectively. This procedure handles either the DG or the TN distributions via its subroutine genUncorrelatedMutation(), which accounts for the type (denoted as {DG, TN}). Calling this subroutine with the DG type obtains a single geometric sample, and hence the need to call it twice and take the difference.
We use $\beta = 0.0873$ rad for the angles' self-adaptation.
For additional parameters' settings see [1].