ISSN 1870-4069

Penalty Functions to Improve the Performance of MOEA's for Portfolio Optimization Problems

Lourdes Uribe, Uriel Trejo-Ramirez,
Yael Andrade-Ibarra, Oliver Cuate,
Victor Cordero

Instituto Politecnico Nacional,
Escuela Superior de Física y Matemáticas,
Mexico

{luriber, ocuateg}@ipn.mx, {utrejor1700, yandradei1600, vcorderoc1600}@alumno.ipn.mx

Abstract. With the significant growth of the financial market, investment options have increased, which can pose a challenge. Thus, one of the most studied problems in the financial field is the Portfolio Optimization Problem, where one seeks the major possible return but with minimal risk. Given that both objectives are in conflict and must be simultaneously optimized, a multi-objective optimization problem (MOP) arises naturally. Even more, since certain conditions must be satisfied, this MOP is restricted; thus, we really are dealing with a constrained MOP (CMOP). Multi-objective evolutionary algorithms (MOEAs) are a widely accepted approach for the numerical treatment of these problems. For constrained problems, however, these methods still have room for improvement to compute satisfactory approximations of the solution sets. In this work, we propose to use different penalty strategies to improve NSGA-II and NSGA-III performance when dealing with the portfolio optimization problem. We claim that penalty strategies helped the evolutionary algorithm to obtain a greater number of feasible individuals while preserving optimal solutions. Numerical results support this claim.

Keywords: Portfolio optimization, penalization, evolutionary algorithms.

1 Introduction

With the significant growth of the financial market, investment options have increased, which can pose a challenge. The availability of numerous options in the market makes it difficult to decide which is the best, even if there is always a single best option. We must remember that every investment comes with risk. If we analyze it carefully, we can identify two different objectives when investing: on the one hand, we aim to maximize investment returns, and on the other hand, we seek to minimize the associated risk.

These objectives often conflict since higher expected returns typically come with higher risks. Such problems are known as multi-objective optimization problems (MOPs). Several approaches have been explored to address these types of problems, commonly involving the application of computational tools.

Algorithm 1 Quadratic Penalty Method.

Require: Given $\mu_0 > 0$, a nonnegative sequence $\{\tau_k\}$ with $\tau_k \to 0$, and a starting point $\boldsymbol{x}_0^s$;
for $k = 0, 1, 2, \ldots$ **do**
  Find an approximate minimizer $\boldsymbol{x}_k$ of Q as in Equation 5, starting at $\boldsymbol{x}_k^s$, and finishing when $\|\nabla Q(\boldsymbol{x})\| \leq \tau_k$;
  if convergence test is satisfied **then**
    stop **return** approximate solution $\boldsymbol{x}_k^s$
  end if
  Choose new penalty parameter $\mu_{k+1} > \mu_k$;
  Choose new starting point $\boldsymbol{x}_{k+1}^s$;
end for

One of these tools is multi-objective algorithms, also known as MOEAs (Multi-Objective Evolutionary Algorithms), which employ techniques inspired by biological evolution to find optimal solutions. MOEAs have caught the interest of many researchers (see, e.g., [8, 5, 3, 9, 12]) over the last decades. Some reasons for this include that MOEAs are of global nature.

Moreover, due to their global approach, they compute a finite size approximation of the entire Pareto Set in one single execution of the algorithm. Also, they have been successfully applied in several applications [18, 20, 29, 24], particularly in the portfolio optimization problem [30, 14].

However, not all of these algorithms handle constraints efficiently. Most MOEAs use feasibility rules to deal with constrained MOPs [7, 15, 21], while others use penalty strategy [25]. Penalty strategy involves assigning a penalty value to infeasible solutions based on the degree of violation. Therefore in the search for optimal solutions, these infeasible solutions will be left behind since they will not get the minimal objective value due to the imposed penalization.

Various families of penalty functions have been studied to improve MOEAs performance when dealing with constrained optimization. There are two main approaches: the first one is based on the constraint violation value, and the second one is based on the distance to the feasible region [13, 25]. One of the most common problems when using both approaches is that the search's effectiveness strongly depends on the selected penalty function.

While evolutionary-guided search with adaptive penalization demonstrates an advantage as an optimization method for these highly restrictive problems [6, 19]. Utilizing feedback from solution search, as well as any specific information, provides an adaptive and dynamic penalization that is effective. The portfolio optimization problem aims to find the optimal distribution of financial assets to maximize expected return and minimize risk. MOEAs provide an effective solution to this problem due to their ability to work with multiple objectives and find optimal points.

The traditional portfolio optimization approach is based on Markowitz's theory. However, this approach assumes normal distributions for asset returns, which can be an idealistic scenario. Additionally, Markowitz's theory does not consider the diversity of objectives that investors may have. Utilization of MOEAs becomes crucial in this context. For example, particle swarm optimization (PSO) has been successfully used in different portfolio optimization problems [30, 14].

Algorithm 2 Classical l_1 Penalty method.

Require: Given $\mu_0 > 0$, tolerance $\tau > 0$ and a starting point $\boldsymbol{x}_0^s$;
  for $k = 0, 1, 2, \ldots$ **do**
    Find an approximate minimizer $\boldsymbol{x}_k^s$ of $\phi_1(x)$, starting at $\boldsymbol{x}_k^s$;
    if $\mathrm{MInf}(\boldsymbol{x}) < \tau$ **then**
      Stop **return** approximate solution $\boldsymbol{x}_k^s$
    end if
    Choose new penalty parameter $\mu_{k+1} > \mu_k$;
    Choose new starting point $\boldsymbol{x}_{k+1}^s$;
  end for

Also, ant colony optimization has been applied to Markowitz's portfolio model [11]. In [2], the authors applied the fireworks algorithm to solve the constrained portfolio problem for the first time. Also, genetic algorithms have been used to solve this problem, specifically in [1] NSGA-II and NSGA-III were employed to solve the portfolio problem for 2 and 3 objectives.

Here, the authors presented that NSGA-II was effective only for two objectives and that NSGA-III was effective only for three objectives. This work aims to optimize various investment portfolios using three different penalty methods implemented on NSGA-II and NSGA-III algorithms.

A comparative analysis is conducted between the results obtained by the MOEA without a penalty and those obtained using the different penalty strategies. Based on the results, we show that when dealing with the portfolio optimization problem, it is very important to implement the correct penalty strategy, as it improves the normal behavior of MOEAs in this problem, especially NSGA-II and NSGA-III.

2 Background

Here, we consider continuous MOPs that can be expressed as:

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^n} \; & \boldsymbol{F}(\boldsymbol{x}), \\ \text{s.t.} \quad & g_i(\boldsymbol{x}) \leq 0 \quad \text{for} \quad i = 1, \ldots, m, \\ & h_i(\boldsymbol{x}) = 0 \quad \text{for} \quad i = 1, \ldots, q. \end{aligned} \tag{1}$$

Hereby, $\boldsymbol{F}$ is the map of objective functions $\boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))^T$. Each objective $f_i : \mathbb{R}^n \to \mathbb{R}$ is assumed for simplicity to be continuously differentiable, and with feasible domain:

$$\Omega = \{\boldsymbol{x} \in \mathbb{R}^n : h_i(\boldsymbol{x}) = 0, i = 1, \ldots, q \text{ and } g_i(\boldsymbol{x}) \leq 0, \ i = 1, \ldots, m\}. \tag{2}$$

The optimality of a MOP is defined using the concept of Pareto dominance: let $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^k$, then $\boldsymbol{v}$ is less or equal than $\boldsymbol{w}$ ($\boldsymbol{v} \leq_p \boldsymbol{w}$), if $v_i \leq w_i$ for all $i \in \{1, \ldots, k\}$; the relation $<_p$ is defined analogously. A vector $\boldsymbol{y} \in \Omega$ is dominated by a vector $\boldsymbol{x} \in \Omega$ ($\boldsymbol{x} \prec \boldsymbol{y}$) with respect to (1) if $\boldsymbol{F}(\boldsymbol{x}) \leq_p \boldsymbol{F}(\boldsymbol{y})$ and $\boldsymbol{F}(\boldsymbol{x}) \neq \boldsymbol{F}(\boldsymbol{y})$, else $\boldsymbol{y}$ is called non-dominated by $\boldsymbol{x}$.

Algorithm 3 Augmented Lagrangian Method.

Require: Given $\mu_0 > 0$, tolerance $\tau > 0$, starting points $\boldsymbol{x}_0^s$ and λ^0;
for $k = 0, 1, 2, \ldots$ **do**
  Find an approximate minimizer $\boldsymbol{x}_k^s$ of $\mathcal{L}_A(\cdot, \lambda^k)$, starting at $\boldsymbol{x}_k^s$, and finishing when $\|\nabla \mathcal{L}_A(\boldsymbol{x}_k; \lambda^k)\| \leq \tau_k$;
  if convergence test is satisfied **then**
    Stop **return** approximate solution $\boldsymbol{x}_k^s$
  end if
  Update Lagrange multipliers using equation 9 to obtain λ^{k+1};
  Choose new penalty parameter $\mu_{k+1} \geq \mu_k$;
  Set starting point for the next iteration to $\boldsymbol{x}_{k+1}^s = \boldsymbol{x}_k$;
  Select tolerance τ_{k+1};
end for

In case $\boldsymbol{F}(\boldsymbol{x}) <_p \boldsymbol{F}(\boldsymbol{y})$ the relation is called strong Pareto dominance. A point $\boldsymbol{x}^* \in \mathbb{R}^n$ is Pareto optimal to (1) if there is no $\boldsymbol{y} \in \Omega$ which dominates $\boldsymbol{x}$. The set of all the Pareto optimal points P_Ω is called the Pareto set, and its image $\boldsymbol{F}(P_\Omega)$ is called the efficient set or Pareto front.

2.1 Portfolio Optimization Problem

The portfolio model, also known as the Markowitz model, aims to maximize the return function while minimizing the risk function. Therefore, a MOP naturally arises. We can define the problem as:

$$\begin{aligned} \text{Max. Return:} \quad & \sum_{i=1}^{N} \boldsymbol{w}_i \mu_i, \\ \text{Min. Risk:} \quad & \sum_{i=1}^{N} \sum_{j=1}^{N} \boldsymbol{w}_i \boldsymbol{w}_j \sigma_{ij}, \\ \text{s.t.} \quad & \sum_{i=1}^{N} \boldsymbol{w}_i = 1, \\ & 0 \leq \boldsymbol{w}_i \leq 1 \quad \text{for} \quad i = 1, \ldots, N, \end{aligned} \tag{3}$$

where N is the number of available assets, μ_i represents the expected return of asset i, σ_{ij} is the covariance between assets i and j, and $\boldsymbol{w}_i$ is the decision variable for asset i. It is worth noticing that w_i has a weighting effect on the return function and the covariance matrix; for more details see [27].

As mentioned before, we try to find solutions that simultaneously satisfy the above conflicting functions. In this work, the optimal portfolio will be the one that provides us with maximum return and minimum risk.

Table 1. This table presents the MOEAs parameters used in the experimental section.

Parameter	NSGAII	NSGAIII
Population size	100	100
Crossover probability	0.9	1
Mutation probability	0.1	1/n
Distribution index for crossover	20	20
Distribution index for mutation	30	20

When diversification is considered, the model can be written as:

$$\begin{aligned}
&\text{Max.} \quad \text{Return:} && \sum_{i=1}^{N} \boldsymbol{w}_i \mu_i - \sum_{i=1}^{N} c_i \mid \boldsymbol{w}_i - \boldsymbol{w}_i^0 \mid, \\
&\text{Min.} \quad \text{Risk:} && \sum_{i=1}^{N} \sum_{j=1}^{N} \boldsymbol{w}_i \boldsymbol{w}_i \sigma_{ij}, \\
&\text{Max.} \quad \text{Entropy:} && \sum_{i=1}^{N} \boldsymbol{w}_i \log(\boldsymbol{w_i}), \\
&\quad \text{s.t.} && \sum_{i=1}^{N} \boldsymbol{w}_i = 1, \\
& && 0 \leq \boldsymbol{w}_i \leq 1 \quad \text{for} \quad i = 1, \ldots, N,
\end{aligned} \tag{4}$$

where $\boldsymbol{w}^0$ is the existing portfolio and $\sum_{i=1}^{N} c_i \mid \boldsymbol{w}_i - \boldsymbol{w}_i^0 \mid$ is the total transaction cost of the portfolio. Here, entropy is used as the divergence measure of asset portfolio in finance literature [17].

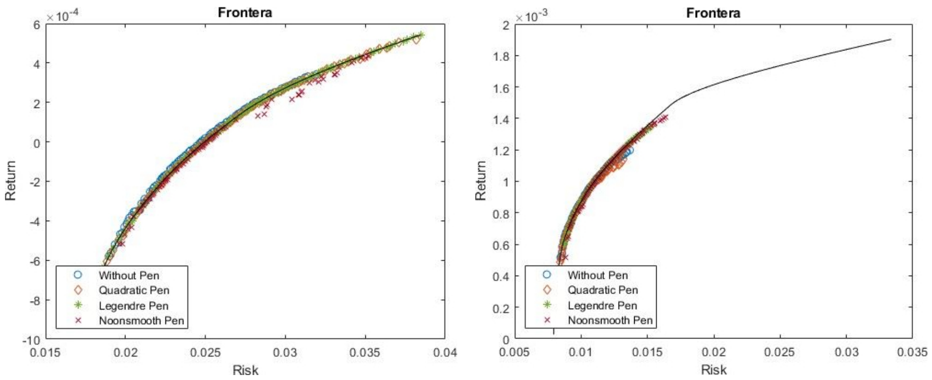
2.2 Penalty Methods

- **Quadratic Penalty Method.** In this method, the penalty terms are the squares of the constraint violations. We define the quadratic penalty function for Problem 1 as:

$$Q(\boldsymbol{x}) = f(\boldsymbol{x}) + \frac{\mu}{2} \sum_{i=1}^{q} h_i^2(\boldsymbol{x}) + \frac{\mu}{2} \sum_{i=1}^{m} (\max\{g_i(\boldsymbol{x}), 0\})^2, \tag{5}$$

  where $\mu > 0$ is the penalty parameter. In Algorithm 1, the general framework based on the quadratic penalty function is presented. It is worth noticing that the parameter sequence $\{\mu_k\}$ can be chosen adaptively, considering the difficulty of minimizing the penalty function at each iteration.

- **Nonsmooth Penalty Function.** Nonsmooth penalty functions are less dependent on the strategy used to choose penalty parameters, which makes them desirable.



(a) (b)

Fig. 1. Comparison of the obtained Pareto fronts for Portfolio 1 and Portfolio 3, respectively, on a certain execution.

A popular nonsmooth penalty function for the general nonlinear programming problem is the l_1 penalty function, which can be defined as:

$$\phi_1(\boldsymbol{x}) = f(\boldsymbol{x}) + \mu \sum_{i=1}^{q} |h_i(\boldsymbol{x})| + \mu \sum_{i=1}^{m} \max\{g_i(\boldsymbol{x}), 0\}, \tag{6}$$

where $\mu > 0$ is the penalty parameter. Note that $\phi_1(x)$ is not differentiable at some $\boldsymbol{x}$ because of the absolute value and $\|\cdot\|$ function. Despite not being differentiable, Equation 6 has a directional derivative along any direction, which allows defining a stationary point of the measure of infeasibility as:

$$\mathrm{MInf}(\boldsymbol{x}) = \sum_{i=1}^{q} |h_i(\boldsymbol{x})| + \sum_{i=1}^{m} \max\{g_i(\boldsymbol{x}), 0\}, \tag{7}$$

When $\mathrm{MInf}(\boldsymbol{x})$ tends to zero, it indicates feasibility. Algorithm 2 presents a general framework based on the l_1 penalty function. Exact nonsmooth penalty functions can be defined in terms of other norms, see [23].

– **Augmented Lagrangian Method: Equality Constraints.** This algorithm is similar to the quadratic penalty algorithm, but it reduces the likelihood of ill-conditioning by introducing Lagrange multipliers into the function. This function is known as the augmented Lagrange function, which preserves smoothness; unlike Nonsmooth penalty functions, the augmented Lagrange function largely preserves smoothness. By definition:

$$\mathcal{L}_A(\boldsymbol{x}, \lambda) = f(x) - \sum_{i=1}^{q} \lambda_i h_i(\boldsymbol{x}) + \frac{\mu}{2} \sum_{i=1}^{q} h_i^2(\boldsymbol{x}), \tag{8}$$

where:

$$\lambda_i^{k+1} = \lambda_i^k - \mu_k h_i(\boldsymbol{x}_k), \; \forall\, i = 1, \ldots, q. \tag{9}$$

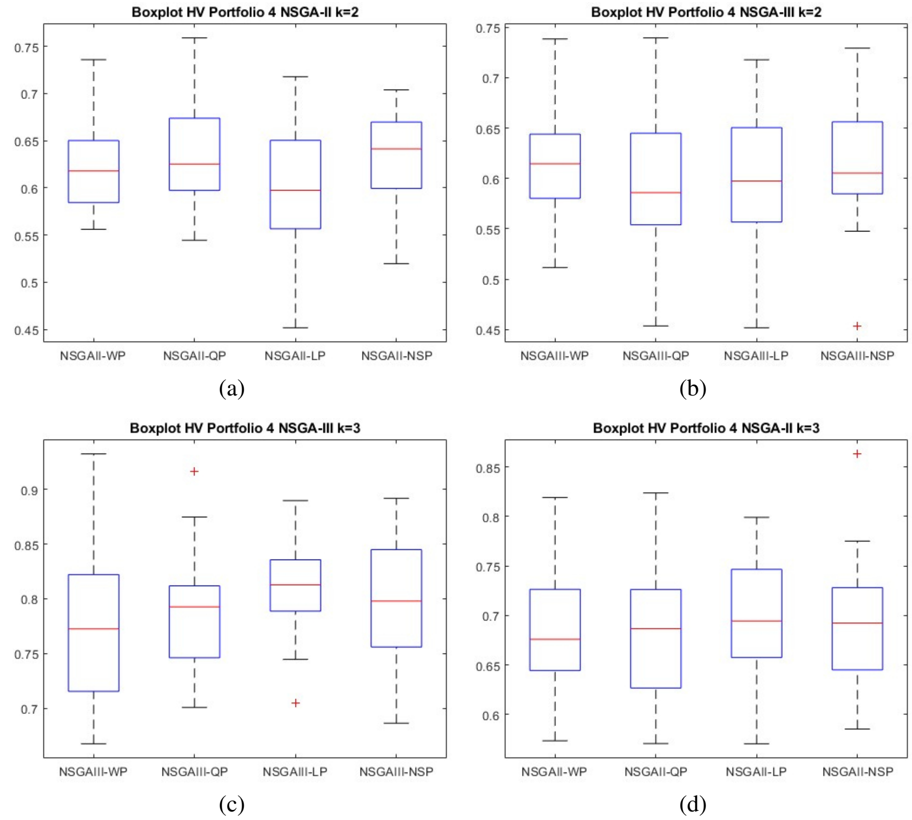


Fig. 2. Boxplots corresponding to HV indicator of Portfolio 4 for two and three objectives.

Notice that Equation 8 only considers equality constraints; thus, inequalities must be transformed. The Augmented Lagrangian Method is presented in Algorithm 3. In this method, the choice of the initial point $\boldsymbol{x}^s_{k+1}$ is less critical when using this method.

3 Proposal

As mentioned in Section 2, MOEAs are useful tools in solving CMOPs. However, these algorithms do not always have a penalty strategy to guide them toward feasible solutions. There are different penalty methods available, in this work, three different methods were employed:

The Quadratic Penalty Method, Nonsmooth Penalty Functions and the Augmented Lagrangian Method [23] to work cooperatively with the selected MOEAs (NSGA-II and NSGA-III [10, 16]). In the following, we present the selected penalty strategies.

3.1 Numerical Results

This section is dedicated to observe the impact that penalty strategies have when used in CMOPs, specifically in the Portfolio Optimization Problem.

Table 2. Average value of the performance indicators of the portfolio problem for $n = 5, 10, 20, 30, 40, 50$ with $k = 2$ via NSGA-II and NSGA-III without penalty strategy (WP), with quadratic penalty (QP), with Nonsmooth penalty (NSP) and with Lagrangian penalty (LP).

NSGA-II												
	FR				Δ_p				**Hv**			
	WP	QP	NSP	LP	WP	QP	NSP	LP	WP	QP	NSP	LP
Portfolio 1	0.9403	0.9883	**1**	0.9917	**1.9185e-04**	7.5454e-04	7.7572e-04	6.5893e-04	0.5728	0.5500	0.5545	**0.6083**
(std.dev)	0.0259	0.0018	0	0.0069	1.8981e-04	1.6311e-04	1.6315e-04	3.5919e-04	2.1016e-04	2.3343e-04	0.0011	0.0029
Portfolio 2	0.8777	0.9423	**0.9997**	0.9460	4.4319e-04	**3.5964e-04**	4.3592e-04	4.7167e-04	0.3327	**0.3535**	0.3321	0.3489
(std.dev)	0.0610	0.0326	0.0018	0.0396	8.7398e-06	2.3808e-05	2.7188e-05	4.0999e-05	0.0878	0.0745	0.0761	0.0995
Portfolio 3	0.6010	**0.8257**	0.7887	0.7710	0.0020	**0.0018**	**0.0018**	**0.0018**	0.6264	0.6310	**0.6445**	0.6199
(std.dev)	0.0660	0.0536	0.0630	0.0541	4.9617e-05	7.3861e-05	5.8708e-05	6.5815e-05	0.0516	0.0.0536	0.0630	0.0541
Portfolio 4	0.8147	**0.9250**	0.9193	0.9227	9.6001e-04	9.5479e-04	**9.0018e-04**	9.7601e-04	0.6233	**0.6360**	0.6352	0.6290
(std.dev)	0.0630	0.0443	0.0370	0.0451	1.1139e-05	1.0845e-05	1.3419e-05	1.0860e-05	0.0537	0.0649	0.0595	0.0626
Portfolio 5	0.8027	0.9087	**0.9240**	0.9230	0.0018	**0.0017**	0.0018	**0.0017**	0.7206	**0.7413**	0.7387	0.7400
(std.dev)	0.0807	0.0537	0.0368	0.0537	2.2405e-05	1.6720e-05	1.6461e-05	1.6641e-05	0.0530	0.0671	0.0628	0.0639
Portfolio 6	0.8083	0.9170	**0.9243**	0.9163	0.0019	**0.0018**	**0.0018**	**0.0018**	0.6907	0.6848	**0.7018**	0.6838
(std.dev)	0.0659	0.0432	0.0362	0.0415	1.5479e-05	1.8858e-05	1.0395e-05	1.8708e-05	0.0543	0.0516	0.0499	0.0512
NSGA-III												
	FR				Δ_p				**Hv**			
	WP	QP	NSP	LP	WP	QP	NSP	LP	WP	QP	NSP	LP
Portfolio 1	0.9743	0.9987	**1**	0.9953	6.6309e-04	**6.0168e-04**	8.5694e-04	0.0013	0.5706	0.5670	0.5551	**0.6528**
(std.dev)	0.0179	0.0035	0	0.0035	1.9198e-04	2.0051e-04	1.3804e-04	3.3193e-04	2.0786e-04	3.3168e-04	5.0220e-04	0.0031
Portfolio 2	0.9490	0.9887	**0.9997**	0.9930	4.6256e-04	**4.3991e-04**	4.7078e-04	6.4490e-04	0.3320	**0.3396**	0.3195	0.3374
(std.dev)	0.0252	0.0063	0.0018	0.0065	7.9075e-06	1.2824e-06	3.5864e-05	2.5302e-05	0.0901	0.0822	0.0989	0.1193
Portfolio 3	0.6853	**0.9283**	0.9033	0.8650	0.0024	0.0023	**0.0022**	0.0023	0.6075	0.6197	0.6283	**0.6286**
(std.dev)	0.1022	0.0385	0.0434	0.0581	6.2940e-05	4.5772e-05	4.0169e-05	4.2092	0.0676	0.0604	0.0713	0.1022
Portfolio 4	0.9010	**0.9627**	0.9623	0.9577	0.0011	0.0011	0.0011	**0.0010**	**0.6183**	0.5958	0.6132	0.6033
(std.dev)	0.0491	0.0215	0.0319	0.0275	9.0502e-06	9.8507e-06	1.9612e-05	1.4121e-05	0.0627	0.0747	0.0632	0.0742
Portfolio 5	0.8717	0.9537	**0.9620**	0.9513	0.0020	**0.0019**	**0.0019**	**0.0019**	0.7249	**0.7341**	0.7320	0.7317
(std.dev)	0.0589	0.0361	0.0277	0.0359	2.0833e-05	2.0766e-05	2.1443e-05	2.0607e-05	0.0604	0.0660	0.0617	0.0668
Portfolio 6	0.8557	0.9457	**0.9517**	0.9460	0.0020	**0.0019**	0.0020	**0.0019**	0.7074	0.6933	**0.7256**	0.6944
(std.dev)	0.0704	0.0332	0.0296	0.0333	2.4282e-05	2.8865e-05	1.6134e-05	2.8940e-05	0.0651	0.0623	0.0472	0.0621

For the numerical experiments, we considered six portfolio problems (for $k = 2$ and $k = 3$, see Equation (3) and Equation (4) respectively), each one related to a different number of assets (5, 10, 20, 30, 40, 50, respectively). We compared the behavior of NSGA-II, NSGA-III, and MOPSO [22] when solving each one of the portfolio problems without penalization (WP) and with different types of penalty strategies (QP, NSP, LP).

For all experiments, we have executed 30 independent runs using 100,000 function evaluations. For the numerical experiments, we used PlatEMO [28]. Table 1 contains the algorithm parameter values used for the experimental setting. The performance indicators Δ_p and $\mathrm{Hypervolume(Hv)}$ [26, 4, 31, 32] are used to measure the penalty strategy effectiveness.

In this work, the real PF used to compute the Δ_p indicator is obtained by theoretically solving the Portfolio Optimization Problem. To compute the Hv indicator, we normalized each objective value of the approximated solution and then set the reference point as $[1,\ 1]$ for two objectives and $[1,\ 1,\ 1]$ for three objectives. We also measure the feasibility rate (FR) of each run. FR is defined as:

$$\mathrm{FR} = \frac{\text{number of feasible individuals}}{\text{number of total individuals}}. \tag{10}$$

Table 3. Average value of the performance indicators of the portfolio problem for $n = 5, 10, 20, 30, 40, 50$ with $k = 3$ via NSGA-II and NSGA-III without penalty strategy (WP), with quadratic penalty (QP), with Nonsmooth penalty (NSP) and with Lagrangian penalty (LP).

NSGA-II								
	FR				**Hv**			
	WP	QP	NSP	LP	WP	QP	NSP	LP
Portfolio 1	0.5386	0.8214	**0.8931**	0.7567	0.5971	0.6329	0.6316	**0.6386**
(std.dev)	0.0320	0.0299	0.0120	0.0824	0.0606	0.0888	0.1225	0.0968
Portfolio 2	0.5070	0.7257	**0.7270**	0.7313	0.5469	**0.5730**	0.5566	0.5618
(std.dev)	0.0426	0.0364	0.0469	0.0450	0.0522	0.0408	0.0368	0.0478
Portfolio 3	0.4437	0.6983	0.6940	**0.7053**	0.7025	**0.7254**	0.7155	0.7179
(std.dev)	0.0491	0.0318	0.0294	0.0487	0.0911	0.0783	0.0982	0.0623
Portfolio 4	0.4227	0.6750	0.6730	**0.6810**	0.6857	0.6815	0.6920	**0.6980**
(std.dev)	0.0498	0.0367	0.0455	0.0370	0.0628	0.0631	0.0623	0.0582
Portfolio 5	0.4150	**0.6770**	0.6757	0.6730	0.6963	0.7007	**0.7065**	0.7047
(std.dev)	0.44	0.0503	0.0398	0.0497	0.0761	0.0701	0.0636	0.0653
Portfolio 6	0.4180	0.6500	**0.6593**	0.6497	0.7272	0.7212	0.7438	**0.7471**
(std.dev)	0.0387	0.0409	0.0486	0.0472	0.0668	0.0598	0.0761	0.0677
NSGA-III								
	FR				**Hv**			
	WP	QP	NSP	LP	WP	QP	NSP	LP
Portfolio 1	0.5994	0.8975	**0.9481**	0.8753	0.7049	0.7291	**0.7471**	0.6936
(std.dev)	0.0399	0.0295	0.0221	0.0905	0.1665	0.1395	0.1521	0.1550
Portfolio 2	0.5642	0.7781	0.7933	**0.7969**	0.6144	0.6213	0.6131	**0.6285**
(std.dev)	0.0473	0.0362	0.0387	0.0633	0.0464	0.0444	0.0346	0.0544
Portfolio 3	0.4772	**0.7283**	0.7083	0.7103	0.7846	0.7916	0.7892	**0.7928**
(std.dev)	0.0531	0.0320	0.0375	0.0460	0.0627	0.0652	0.0481	0.0571
Portfolio 4	0.4150	0.6847	**0.6944**	0.6733	0.7750	0.7873	0.7951	**0.8106**
(std.dev)	0.0488	0.0425	0.0349	0.0386	0.0644	0.0508	0.0532	0.0425
Portfolio 5	0.4008	0.6628	0.6636	**0.6719**	0.7890	**0.8083**	0.7985	0.7962
(std.dev)	0.0436	0.0377	0.0527	0.0376	0.0592	0.0551	0.0499	0.0620
Portfolio 6	0.3861	0.6536	**0.6578**	0.6450	0.7819	0.8016	**0.8194**	0.7946
(std.dev)	0.0370	0.0495	0.0459	0.0461	0.0539	0.0610	0.0607	0.0542

We claim that by using penalty strategies, not only does the feasibility rate improve, but we also improve the performance of the MOEAs when solving the portfolio optimization problem. Table 2 shows the obtained results using NSGA-II and NSGA-III for the portfolio problem of $k = 2$ and Table 3 shows the obtained results using NSGA-II and NSGA-III for the portfolio problem of $k = 3$.

MOPSO algorithm had troubles when solving the selected CMOPs. When the number of assets increases ($n > 10$), the algorithm fails in finding feasible solutions. Figure 1 shows the behavior of NSGA-II with and without penalty strategy on a certain execution for Portfolio 1 and 3.

Observe that more feasible solutions can be obtained by implementing a penalty function. First, recall that the FR indicator measures feasibility. If the indicator value tends to 1, there is a higher prevalence of feasible solutions. Note that in all portfolios, the FR indicator is always higher when a penalty function is applied. Additionally, the WP value is always the smallest, meaning NSGA will always obtain more feasible solutions by incorporating a penalty strategy.

Now, the Δp indicator aims for convergence and distribution; a smaller value indicates higher performance. Analyzing the case of $k = 2$, the first portfolio, we notice that the Δp indicator without applying penalty functions is lower than when a penalty function is applied; in this case, we are considering only five decision variables. Therefore, the standalone NSGA-II is enough to solve the problem.

However, in the remaining five portfolios (more variables), this indicator is always better when some penalty function is applied. Finally, we have the Hv indicator, which measures the volume of the space dominated by a set of solutions in the objective space, so this indicator should tend to 1 when all objectives are normalized.

Note that the Hv indicator is higher in all portfolios when some penalty function is applied. Note that for NSGA-III, we have a similar behavior. In all portfolios, the FR indicator is always higher when some penalty function is applied, and the value of the WP indicator is always the smallest. Also, the penalty versions outbeat the standalone algorithm referring to Δ_p and Hv indicators. Only in Portfolio 3 the higher value of HV is found in WP.

Finally, considering the portfolio problem for $k = 3$ one can notice that, as expected, penalty strategies helped the evolutionary framework and obtained solutions of higher quality. Although the FR value is no longer as good as for $k = 2$, it still is better than the standalone version.

For this case, we only measure the Hv indicator since we do not know the real Pareto front of the problem. Finally, boxplots corresponding to the Portfolio 4 problem for two and three objectives using the Hv indicator are presented in Figure 2. Observe that the MOEA version with a penalty strategy gets better results than the standalone algorithm.

4 Conclusions

In this work, we deal with the Portfolio Optimization Problem; since it can be defined as a CMOP, we are interested in analyzing how well it can be solved by MOEAs considering penalty strategies. Our proposal uses three different penalty functions, each with a specific characteristic. We claim that better performance is expected when a MOEA employs a penalty strategy. Several numerical results support this claim.

Numerical experiments showed that penalty strategies helped the behavior of the evolutionary framework, not only in terms of performance indicators (Δ_p and Hv) but also by obtaining a major number of feasible individuals. Since it is a constrained optimization problem, we aim for feasibility and optimality. Although we have promising results, it is worth noticing that these are preliminary results since some aspects are still pending exploration. For example, we focused here on the most common portfolio optimization problem (2 and 3 objectives), but this problem can be more difficult if more constraints are considered, as the ones used in [2].

Our next steps include studying different types of portfolios and analyzing which penalization strategy is more suitable for these types of MOPs, aiming for theoretical results that back up them.

Acknowledgments. Lourdes Uribe, Uriel Trejo-Ramirez and Yael Andrade-Ibarra acknowledge support from project no. SIP20232208. Oliver Cuate and Victor Cordero acknowledge support from project no. SIP20221947 and IPN-SIP 20231045.

References

1. Awad, M., Abouhawwash, M., Agiza, N. H.: On NSGA-II and NSGA-III in portfolio management. Intelligent Automation and Soft Computing, vol. 32, no. 3, pp. 1893–1904 (2022) doi: 10.32604/iasc.2022.023510
2. Bacanin, N., Tuba, M.: Fireworks algorithm applied to constrained portfolio optimization problem. In: IEEE Congress on Evolutionary Computation, pp. 1242–1249 (2015) doi: 10.1109/CEC.2015.7257031
3. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research, vol. 181, no. 3, pp. 1653–1669 (2007) doi: 10.1016/j.ejor.2006.08.008
4. Bogoya, J. M., Vargas, A., Schütze, O.: The averaged Hausdorff distances in multi-objective optimization: A review. Mathematics Multidisciplinary Digital Publishing Institute, vol. 7, no. 10, pp. 1–35 (2019) doi: 10.3390/math7100894
5. Coello-Coello, C. A., Lamont, G. B., Van-Veldhuizen, D. A.: Evolutionary algorithms for solving multi-objective problems. Springer (2007)
6. Coit, D. W., Smith, A. E., Tate, D. M.: Adaptive penalty methods for genetic optimization of constrained combinatorial problems. Institute for Operations Research and the Management Sciences Journal on Computing, vol. 8, no. 2, pp. 173–182 (1996) doi: 10.1287/ijoc.8.2.173
7. Deb, K.: An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering, vol. 186, no. 2–4, pp. 311–338 (2000) doi: 10.1016/s0045-7825(99)00389-8
8. Deb, K.: Multi-objective optimization using evolutionary algorithms: An Introduction. John Wiley and Sons, Inc (2011) doi: 10.1007/978-0-85729-652-8_1
9. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation, vol. 18, no. 4, pp. 577–601 (2014) doi: 10.1109/TEVC.2013.2281535
10. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182–197 (2002) doi: 10.1109/4235.996017
11. Deng, G. F., Lin, W. T.: Ant colony optimization for Markowitz mean-variance portfolio model. In: Swarm, Evolutionary, and Memetic Computing: First International Conference on Swarm, Evolutionary, and Memetic Computing, vol. 6466, pp. 238–245 (2010) doi: 10.1007/978-3-642-17563-3_29
12. Fan, Z., Li, W., Cai, X., Li, H., Wei, C., Zhang, Q., Deb, K., Goodman, E.: Push and pull search for solving constrained multi-objective optimization problems. Swarm and evolutionary computation, vol. 44, pp. 665–679 (2019) doi: 10.1016/j.swevo.2018.08.017
13. Goldberg, D. E.: Genetic algorithms in search, optimization and machine learning. ACM Digital Library (1989)

14. Golmakani, H. R., Fazel, M.: Constrained portfolio selection using particle swarm optimization. Expert Systems with Applications, vol. 38, no. 7, pp. 8327–8335 (2011) doi: 10.1016/j.eswa.2011.01.020
15. He, Q., Wang, L.: A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. Applied Mathematics and Computation, vol. 186, no. 2, pp. 1407–1422 (2007) doi: 10.1016/j.amc.2006.07.134
16. Jain, H., Deb, K.: An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach. IEEE Transactions on Evolutionary Computation, vol. 18, no. 4, pp. 602–622 (2014) doi: 10.1109/tevc.2013.2281534
17. Jana, P., Roy, T. K., Mazumder, S. K.: Multi-objective possibilistic model for portfolio selection with transaction cost. Journal of Computational and Applied Mathematics, vol. 228, no. 1, pp. 188–196 (2009) doi: 10.1016/j.cam.2008.09.008
18. Kamal-Abasi, A., Tajudin-Khader, A., Al-Betar, M. A., Naim, S., Makhadmeh, S. N., Alkareem-Alyasseri, Z. A.: Link-based multi-verse optimizer for text documents clustering. Applied Soft Computing, vol. 87, pp. 106002 (2020) doi: 10.1016/j.asoc.2019.106002
19. Knypiński, Ł.: Adaptation of the penalty function method to genetic algorithm in electromagnetic devices designing. The International Journal for Computation and Mathematics in Electrical and Electronic Engineering, vol. 38, no. 4, pp. 1285–1294 (2019) doi: 10.1108/COMPEL-01-2019-0010
20. Lu, H., Chen, J., Yan, K., Jin, Q., Xue, Y., Gao, Z.: A hybrid feature selection algorithm for gene expression data classification. Neurocomputing, vol. 256, pp. 56–62 (2017) doi: 10.1016/j.neucom.2016.07.080
21. Mezura-Montes, E., Coello-Coello, C. A., Tun-Morales, E. I.: Simple feasibility rules and differential evolution for constrained optimization. In: Mexican International Conference on Artificial Intelligence, pp. 707–716 (2004) doi: 10.1007/978-3-540-24694-7_73
22. Moore, J., Chapman, R.: Application of particle swarm to multiobjective optimization. Department of Computec Science and Software Engineering, Anburn University (1999)
23. Nocedal, J., Wright, S. J.: Numerical optimization. Springer (1999)
24. Ravizza, S., Chen, J., Atkin, J. A. D., Burke, E. K., Stewart, P.: The trade-off between taxi time and fuel consumption in airport ground movement. Public Transport, vol. 5, no. 1–2, pp. 25–40 (2013) doi: 10.1007/s12469-013-0060-1
25. Richardson, J. T., Palmer, M. R., Liepins, G. E., Hilliard, M. R.: Some guidelines for genetic algorithms with penalty functions. In: Proceedings of the 3rd International Conference on Genetic Algorithms, pp. 191–197 (1989)
26. Schutze, O., Esquivel, X., Lara, A., Coello-Coello, C. A.: Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization. IEEE Transactions on Evolutionary Computation, vol. 16, no. 4, pp. 504–522 (2012) doi: 10.1 109/tevc.2011.2161872
27. Sefiane, S., Benbouziane, M.: Portfolio selection using genetic algorithm. Journal of Applied Finance and Banking, vol. 2, no. 4, pp. 143–154 (2012)
28. Tian, Y., Cheng, R., Zhang, X., Jin, Y.: PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. IEEE Computational Intelligence Magazine, vol. 12, no. 4, pp. 73–87 (2017) doi: 10.1109/mci.2017.2742868
29. Zhang, K., Du, H., Feldman, M. W.: Maximizing influence in a social network: Improved results using a genetic algorithm. Physica A: Statistical Mechanics and its Applications, vol. 478, pp. 20–30 (2017) doi: 10.1016/j.physa.2017.02.067
30. Zhu, H., Wang, Y., Wang, K., Chen, Y.: Particle swarm optimization (PSO) for the constrained portfolio optimization problem. Expert Systems with Applications, vol. 38, no. 8, pp. 10161–10169 (2011) doi: 10.1016/j.eswa.2011.02.075

31. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms - a comparative case study. In: Parallel Problem Solving from Nature, pp. 292–301 (1998) doi: 10.1007/bfb0056872
32. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., da Fonseca, V. G.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Transactions on Evolutionary Computation, vol. 7, no. 2, pp. 117–132 (2003) doi: 10.1109/tevc.2003.810758