Moroccan J. of Pure and Appl. Anal. (MJPA)A)

Volume 7(2), 2021, Pages 299–311

ISSN: Online 2351-8227 - Print 2605-6364

DOI: [10.2478/mjpaa-2021-0019](https://doi.org/10.2478/mjpaa-2021-0019)

Combining the cross-entropy algorithm and ϵ -constraint method for multiobjective optimization

ABDELMAJID EZZINE¹, ABDELLAH ALLA², AND NADIA RAISSI³

ABSTRACT. This paper aims to propose a new hybrid approach for solving multiobjective optimization problems. This approach is based on a combination of global and local search procedures.

The cross-entropy method is used as a stochastic model-based method to solve the multiobjective optimization problem and reach a first elite set of global solutions. In the local search step, an ϵ -constraint method converts the multiobjective optimization problem to a series of parameterized single-objective optimization problems. Then, sequential quadratic programming (SQP) is used to solve the derived single-objective optimization problems allowing to reinforce and improve the global results.

Numerical examples are used to demonstrate the efficiency and effectiveness of the proposed approach.

Mathematics Subject Classification (2020). 90C29, 90C15, 90C30.

Key words and phrases. Multiobjective optimization, Cross-entropy method, ϵ -constraint method, SQP methods.

1. Introduction

The interest in multiobjective optimization has grown enormously in recent decades. It has been mainly focused on many areas of science and engineering, such as portfolio management [1], distribution of energy [2], aerodynamic design [3], and so on [26, 27, 28].

Received : August 31, 2020 - Accepted: January 02, 2021.

© The Author(s) 2020. This article is published with open access by Sidi Mohamed Ben Abdallah University.

^{1,2,3} *Laboratory of Analysis Mathematics and Applications, Department of Mathematics, Faculty of Sciences, Mohammed V University in Rabat, Morocco.*

*e-mail*¹: abdelmajid.ezzine@um5.ac.ma (Corresponding Author),

*e-mail*²: abdellah.alla@um5.ac.ma,

*e-mail*³: nadia.raissi@um5.ac.ma.

Unlike single-objective optimization, the primary motivation for multiobjective optimization is to satisfy and optimize several conflicting objectives simultaneously. However, the concept of optimality of solutions is replaced by the concept of Pareto optimality. Where efficient solutions (also called non-dominated solutions) are selected via the Pareto non-dominance process [4, 5].

There are different methods and approaches for solving multiobjective optimization problems. According to [6] and [7], we can cite two types of methods to generate the set of Pareto optimal solutions. Scalarization methods, which convert the multiobjective optimization problem to several parametric single-objective optimization problems. The second approach is vectorization methods, which are often population-based methods and which directly use the multiobjective structure of the problem.

In this article, a new hybrid approach is proposed, which combines the two approaches to solving multiobjective optimization problems (MOP) mentioned above. The resulting algorithm follows two directions of research, global and local research to reach efficient solutions.

The global search is based on the cross-entropy method (CE). It is a population-based stochastic method for optimization and importance sampling, and which minimizes the Kullback-Liebler divergence (or cross-entropy) between two probability distributions associated with a sample of solutions to generate a better-optimized sample. It was initially presented by Rubinstein and Kroese [10, 11], and it is originally motivated by the simulation of rare events [12].

Concerning the local research, we consider the ϵ -constraint method, which consists of choosing at each decomposition of the objective space, an objective function to be optimized with high priority, and to transform the remaining objective functions into inequality constraints bounded by an initial constraint vector [6]. The single objective problem derived will be solved by one of the sequential quadratic programming (SQP) approaches of literature in [13, 15, 14].

The content of this paper is organized as follows: The first section consists of some basic definitions and concepts of multiobjective optimization. The second section explains our proposed hybrid approach and details the implementation of the resulting algorithm. In the third section, we show some numerical results to demonstrate the efficiency and competitiveness of our approach. Finally, we present conclusions and introduce ideas for future work related to this paper.

2. Basic Concepts and Notations

We consider the multiobjective optimization problem (MOP) formulated as follows:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & (f_1(x), f_2(x), \dots, f_M(x)) \\ \text{subject to} \quad & g_p(x) = 0 \quad p \in \{1, \dots, P\} \\ & h_q(x) \leq 0 \quad q \in \{1, \dots, Q\} \end{aligned} \tag{2.1}$$

where $x = (x_1, \dots, x_n)$ is the vector of decision variable, $f_m : \mathbb{R}^n \rightarrow \mathbb{R}$, $m \in \{1, \dots, M\}$ are the objective functions, $g_p : \mathbb{R}^n \rightarrow \mathbb{R}$, $p \in \{1, \dots, P\}$ are the inequality constraints and $h_q : \mathbb{R}^n \rightarrow \mathbb{R}$, $q \in \{1, \dots, Q\}$ are the equality constraints.

We assume that

$$x_i^L \leq x_i \leq x_i^U \quad \forall i \in \{1, \dots, n\}.$$

where x_i^L and x_i^U are respectively the lower and the upper bound of x_i . The bound constraints associated to x_i are assumed to be explicit in the inequality constraints.

Moreover, we assume also that the objective functions and the constraints functions are twice continuously differentiable.

We state and discuss some definitions and notations used to describe the Pareto optimal solutions of a multiobjective optimization problem (MOP).

Definition 2.1. Let $x_u, x_v \in X \subset \mathbb{R}^n$ two vectors of decision variables,

- We say that x_u weakly dominates x_v (denoted $x_u \leq x_v$), if and only if :

$$\forall m \in \{1, \dots, M\} \quad f_m(x_u) \leq f_m(x_v);$$

- We say that x_u dominates x_v (denoted $x_u < x_v$), if and only if, x_u weakly dominates x_v and

$$\exists m_0 \in \{1, \dots, M\} \quad f_{m_0}(x_u) < f_{m_0}(x_v).$$

Definition 2.2. Let $x^* \in \mathcal{F}$ ($\mathcal{F}$ is the feasible region) a vector of decision variables.

- We say that x^* is Weakly Pareto optimal if there does not exist another $x \in \mathcal{F}$ such that

$$f_m(x) < f_m(x^*) \quad \forall m \in \{1, \dots, M\}$$

- We say that x^* is Pareto optimal if there does not exist another $x \in \mathcal{F}$ such that

$$\begin{aligned} \forall m \in \{1, \dots, M\} \quad & f_m(x) \leq f_m(x^*); \\ \exists m_0 \in \{1, \dots, M\} \quad & f_{m_0}(x) < f_{m_0}(x^*) \end{aligned}$$

Definition 2.3. We say that $\mathcal{P}^*$ is the Pareto optimal set (or the efficient set), if $\mathcal{P}^*$ is the non dominated set of the entire feasible decision space X , and it is defined as:

$$\mathcal{P}^* = \{x \in X / \nexists u \in X : f(u) < f(x)\}. \quad (2.2)$$

Definition 2.4. The Pareto front $\mathcal{P}_{\mathcal{F}}^*$ associated to a Pareto optimal set $\mathcal{P}^*$ is defined as :

$$\mathcal{P}_{\mathcal{F}}^* = \{f(x) \in \mathbb{R}^n / x \in \mathcal{P}^*\}. \quad (2.3)$$

3. The Proposed Method for Solving Multiobjective Optimization Problems

In this section, we state and describe the proposed hybrid approach, as well as the implementation of its resulting algorithm.

3.1. The Global Search Step. In this step, we use the previously introduced cross-entropy method (CE). First, we will apply this method to a problem derived from the initial problem and augmented with a penalty function ϕ . The problem raised without constraints is established as follows:

$$\min_{x \in \mathbb{R}^n} \hat{f}(x) = (f_1(x), f_2(x), \dots, f_m(x), \phi(x)) \quad (3.1)$$

The penalty function ϕ is included as an objective function of the initial problem. It is defined by:

$$\phi(x) = \sum_{p=1}^P [g_p(x)]^+ + \sum_{q=1}^Q |h_q(x)| \quad (3.2)$$

where

$$[g(x)]^+ = \begin{cases} g(x) & \text{if } g(x) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.3)$$

The cross-entropy method (CE) consists in generating a sample of admissible random solutions $X_1, \dots, X_N$ at each iteration t according to a parameterized probability density. In our case, we use the multidimensional normal distribution $\mathcal{N}(\mu_t, \sigma_t^2)$. Inspired by [8], we set the initial parameters μ_0 and σ_0^2 as follows:

For all $i \in \{1, \dots, n\}$,

$$\begin{aligned} \mu_{0,i} &= (x_i^L + x_i^U)/2; \\ \sigma_{0,i} &= \mu_{0,i} - x_i^L \end{aligned} \quad (3.4)$$

As indicated in [9] by minimizing the cross-entropy (or the Kullback-Liebler distance), updating the parameters of the sample distribution on the basis of an efficient selected sample of solutions, allows to converge towards an optimal solution.

Let N_t^* be the length of the non dominated sample of solutions $\mathcal{P}_t^*$ selected at the iteration t and $\mathcal{I}$ the index of the efficient solutions selected, the update of the parameters μ_t and σ_t^2 is given as follows:

$$\begin{aligned} \mu_t &= \gamma \tilde{\mu}_t + (1 - \gamma) \mu_{t-1}; \\ \sigma_t &= \gamma \tilde{\sigma}_t + (1 - \gamma) \sigma_{t-1}. \end{aligned} \quad (3.5)$$

With γ is a smoothing parameter and $\tilde{\mu}_t, \tilde{\sigma}_t$ are the mean and standard deviation of the selected efficient sample define as:

For all $i \in \{1, \dots, n\}$,

$$\begin{aligned} \tilde{\mu}_{t,i} &= \sum_{k \in \mathcal{I}} x_{k,i} / N_t^*; \\ \tilde{\sigma}_{t,i}^2 &= \sum_{k \in \mathcal{I}} (x_{k,i} - \tilde{\mu}_{t,i})^2 / N_t^*. \end{aligned} \quad (3.6)$$

The performing sample $\mathcal{P}_t^*$ converges to the Pareto optimal set of solutions of the multiobjective optimization problem (MOP) until $\tilde{\sigma}_{t,j}$ reaches a sufficiently small tolerance parameter ϵ_{tol} .

The cross-entropy algorithm is then stated as:

Algorithm 1 The cross-entropy algorithm

Step 1: Choose $\mu_0(x_0)$ and σ_0 according to (3.4).

Set iterator $t = 1$.

Step 2: Generate a random sample $X_1, \dots, X_N$ from the multivariate normal distribution $\mathcal{N}(\mu_{t-1}, \sigma_{t-1}^2)$.

Step 3: Let $\mathcal{P}_t^*$ the performing samples according to Pareto Optimality and $\mathcal{I}$ be the indices of the performing samples $\mathcal{P}_t^*$.

Step 4: Update $\tilde{\mu}_t$ and $\tilde{\sigma}_t$ using (3.6).

Step 5: Smooth μ_t and σ_t using (3.5).

Step 6: If $\max_j(\tilde{\sigma}_{t,j}) < \epsilon_{tol}$ (ϵ_{tol} a tolerance parameter) **then** stop and return $\mathcal{P}_t^*$ as the Pareto Optimal set of solutions to the MOP. **Otherwise**, increase t by 1 and return to **Step 2**.

3.2. The Local Search Step. In this part, we will describe the components of the local search procedure. This step consists in locally improving the results of global research. More specifically, new efficient solutions will be generated from a local search around the sample of the efficient solutions $\mathcal{P}_t^*$ previously generated in the global search. Despite the time consuming by the local optimization, this makes it possible to reach critical regions for global research. A precision threshold will be adopted to control the search direction in the objective space, and thus optimize the execution time.

First, we implement the ϵ -constraint method to the initial problem. This method transforms the multiobjective optimization problem into a series of parameterized single-objective problems. It consists of choosing an objective function of high priority to be optimized and initial constraint values. And It turns the rest of the objective functions into inequality constraints.

Let j_0 be the index of the objective function chosen, and $\epsilon_j \geq 0, \forall j \neq j_0, j \in \{1, \dots, M\}$ the initial constraints values. Therefore, the initial problem is converted into a series of single-objective problems of the form:

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} f_{j_0}(x) \\ & \text{subject to} \quad f_j(x) \leq \epsilon_j \quad \forall j \neq j_0 \quad j \in \{1, \dots, M\} \\ & \quad \quad \quad g_p(x) \leq 0 \quad p \in \{1, \dots, P\} \\ & \quad \quad \quad h_q(x) = 0 \quad q \in \{1, \dots, Q\} \end{aligned} \tag{3.7}$$

We define the set of couples of efficient solutions $\mathcal{P}_t^*$ in the following way:

$$\begin{aligned} V_j = \{ (x^a, x^b) \in \mathcal{P}_t^* \times \mathcal{P}_t^* / \forall x^c \in \mathcal{P}_t^* \setminus \{x^a, x^b\}; \\ f_j(x^c) \leq \min_{x \in \{x^a, x^b\}} f_j(x) \text{ or } f_j(x^c) \geq \max_{x \in \{x^a, x^b\}} f_j(x) \} \end{aligned} \tag{3.8}$$

The initial constraint values ϵ_j are chosen such that the minimum distance between efficient solutions around ϵ_j with respect to an objective function is greater than a given precision threshold η .

$$\epsilon_j \in \tilde{\zeta}_j \quad (3.9)$$

with

$$\tilde{\zeta}_j = \{y \in \mathbb{R} / \exists (x^a, x^b) \in V_j; |f_j(x^a) - f_j(x^b)| > \eta \quad \text{and} \quad f_j(x^a) \leq y \leq f_j(x^b)\} \quad (3.10)$$

Since all the components of the single-objective problem are well defined, the set of possible combinations of $\tilde{\zeta}_j \forall j \neq j_0 \ j \in \{1, \dots, M\}$ defines the sequence of parametric single-objective problems to be resolved according to specific regions in the objective space.

Inspired by [20], we choose to solve the single-objective problems using one of the sequential quadratic programming (SQP) approaches.

In our case, we use the implementation of [14] to solve the derived single-objective problems. The convergence of this type of method is discussed in [15].

3.3. The resulting CEεC algorithm. Our resulting algorithm differs from the classical cross-entropy algorithm described above (3.1) by integrating a deterministic local search just before updating the parameters of the sample distribution of the global search.

The Cross-Entropy-ε-Constraint algorithm proposed is then stated as follows:

Algorithm 2 The CEεC algorithm

Step 1: Choose $\mu_0(x_0)$ and σ_0 according to (3.4).

Set iterator $t = 1$.

Step 2: Generate a random sample $X_1, \dots, X_N$ from the multivariate normal distribution $\mathcal{N}(\mu_{t-1}, \sigma_{t-1}^2)$.

Step 3: Let $\mathcal{P}_t^*$ the performing samples according to Pareto Optimality.

Step 4: Solve the Local search problem (3.7) for each $x \in \mathcal{P}_t^*$ and $\epsilon_j \in \tilde{\zeta}_j$ according to (3.10) and get $\mathcal{P}_t^{**}$ the improved non dominated set. Let $\mathcal{I}$ be the indices of the improved performing samples $\mathcal{P}_t^{**}$.

Step 5: Update $\tilde{\mu}_t$ and $\tilde{\sigma}_t$ using (3.6).

Step 6: Smooth μ_t and σ_t using (3.5).

Step 7: If $\max_j(\tilde{\sigma}_{t,j}) < \epsilon_{tol}$ **then** stop and return $\mathcal{P}_t^{**}$ as the Pareto Optimal set of solutions to the MOP. **Otherwise**, increase t by 1 and return to **Step 2**.

4. Numerical Results and Discussion

We are now evaluating the performance of the proposed algorithm for solving multiobjective optimization problems. The algorithm was coded in MATLAB and tested on a PC with 2.8 GHz Intel CPU I7. The computational experiments were carried out on four reference problems

taken from [17, 18, 24, 25]. The chosen benchmark test problems are given in Table 1 as follows:

MOP	Objective functions	Constraints
Binh2	$f_1(x) = 4x_1^2 + 4x_2^2$	$0 \leq x_1 \leq 5$
	$f_2(x) = (x_1 - 5)^2 + (x_2 - 5)^2$	$0 \leq x_2 \leq 3$
		$(x_1 - 5)^2 + x_2^2 - 25 \leq 0$
		$-(x_1 - 8)^2 - (x_2 + 3)^2 + 7.7 \leq 0$
Tanaka	$f_1(x) = x_1$	$0 \leq x_1 \leq \pi$
	$f_2(x) = x_2$	$0 \leq x_2 \leq \pi$
		$-x_1^2 - x_2^2 + 1$
		$+ \frac{\cos(16\arctan(x_1/x_2))}{10} \leq 0$
Srinivas	$f_1(x) = (x_1 - 2)^2 + (x_2 - 1)^2 + 2$	$-20 \leq x_1 \leq 20$
	$f_2(x) = 9x_1 - (x_2 - 1)^2$	$-20 \leq x_2 \leq 20$
		$x_1^2 + x_2^2 - 225 \leq 0$
		$x_1^2 - 3x_2^2 + 10 \leq 0$
ZDT3	$f_1(x) = x_1$	$0 \leq x_i \leq 1 \ \forall i \in \{1, \dots, n\}$
	$f_2(x) = \alpha(x) \left(1 - \sqrt{\frac{x_1}{\alpha(x)}} - \frac{x_1}{\alpha(x)} \sin(10\pi x_1) \right)$	
	with $\alpha(x) = 1 + \frac{9}{n-1} \cdot \sum_{i=2}^n x_i$	

TABLE 1. The multiobjective test problems

After some experimentation, we opted for the parameters below 2 for the numerical simulation of the benchmark test problems 1. The parameters chosen are defined for each test problem and each level of research for Pareto optimal solutions (global research and local research parameters). The choice obviously depends on the decision space and the number of objectives of each problem, and more particularly, on the extent of the true Pareto front associated to the problem.

For illustration, Figures (1),(2),(3) and (4) show the computational results of the efficient set of solutions (in red), where the evolution of sample generation during the algorithm process is marked in blue. We notice that the spread between generated solutions decreases progressively and gradually until convergence to non-dominant solutions (efficient solutions).

Figures (1) and (3) presenting numerical simulations of Binh and Tanaka problems have convex and connected Pareto optimal fronts. While the Pareto optimal fronts of the Tanaka and ZDT3 problems shown in figures (2) and (4) respectively are disconnected.

To evaluate the convergence of the proposed algorithm, we opted for convergence metrics in order to demonstrate the performance of the algorithm and its competitiveness vis--vis other approaches to literature.

We discuss two convergence metrics as follows:

MOP	Dimensions	Global Search parameters	Local Search parameters
Binh2	$n = 2$	$N = 2500$	$\eta = 0.1$
	$m = 2$	$\gamma = 0.5$	
		$\epsilon_{tol} = 0.001$	
		$T_{max} = 10$	
Tanaka	$n = 2$	$N = 100$	$\eta = 0.01$
	$m = 2$	$\gamma = 0.5$	
		$\epsilon_{tol} = 0.001$	
		$T_{max} = 10$	
Srinivas	$n = 2$	$N = 100$	$\eta = 1$
	$m = 2$	$\gamma = 0.5$	
		$\epsilon_{tol} = 0.001$	
		$T_{max} = 10$	
ZDT3	$n = 30$	$N = 100$	$\eta = 0.1$
	$m = 2$	$\gamma = 0.5$	
		$\epsilon_{tol} = 0.01$	
		$T_{max} = 500$	

TABLE 2. Parameters

- **Generational distance GD:** This metric allows to measure the Euclidean distance between each solution and the nearest neighbors of the true Pareto front. It is stated as follows:

$$GD = \frac{1}{|\mathcal{P}_{\mathcal{F}}|} \sqrt{\sum_{i=1}^{|\mathcal{P}_{\mathcal{F}}|} d_i^2} \quad (4.1)$$

where $|\mathcal{P}_{\mathcal{F}}|$ is the dimension of the obtained Pareto front and d_i is the Euclidean distance between each element of the optimal Pareto front obtained $\mathcal{P}_{\mathcal{F}}$ and the closest element of the True Pareto front of the problem.

- **Diversity Metric Δ :** This metric allows to measure the extent of spread reached among the obtained solutions $\mathcal{P}_{\mathcal{F}}$. It follows as:

$$\Delta = \frac{d_a + d_b + \sum_{i=1}^{|\mathcal{P}_{\mathcal{F}}|-1} |d_i - \bar{d}|}{d_a + d_b + (|\mathcal{P}_{\mathcal{F}}| - 1)\bar{d}} \quad (4.2)$$

where d_i is the Euclidean distance between successive solutions in the Pareto optimal set, $\bar{d}$ is the average distance, d_a and d_b are the Euclidean distances between the boundary of the obtained solutions and the corresponding extreme solutions.

FIGURE 1. Representation of the Pareto optimal front of the Binh test problem (red) and the evolution of sample generation (blue).

FIGURE 2. Representation of the Pareto optimal front of the Tanaka test problem (red) and the evolution of sample generation (blue).

We compare our algorithm results to two evolutionary multiobjective optimization algorithms, The non dominated sorting genetic algorithm II (NSGA-II) [21] and the multiobjective

FIGURE 3. Representation of the Pareto optimal front of the Srinivas test problem (red) and the evolution of sample generation (blue).

FIGURE 4. Representation of the Pareto optimal front of the ZDT3 test problem (red) and the evolution of sample generation (blue).

particle swarm-differential evolution algorithm with a differential evolution (MOPSD) [22].

Both of these algorithms have been studied on different benchmark test problems that contain the test problem (1) chosen in this comparison.

The results of the convergence metrics for the reference problems are:

MOP	GD			Δ		
	NSGA-II	MOPSDE	CE ϵ C	NSGA-II	MOPSDE	CE ϵ C
Binh2	1.0075	0.0708	0.0034	1.4822	0.7734	0.9512
Tanaka	0.0040	0.0007	0.0005	0.8228	0.8594	0.9548
Srinivas	0.0037	0.8643	0.0489	0.3869	0.6525	0.7882
ZDT3	0.0024	0.0006	0.0055	0.59025	0.5817	1.1831

TABLE 3. GD and Δ metrics obtained by NSGA-II, MOPSDE and CE ϵ C algorithms on the benchmark test problems.

From Table (3) we show that the GD obtained by CE ϵ C is smaller than those achieved by NSGA-II and MOPSDE for both problems Binh2 and Tanaka, which indicate that the convergence of solutions reached by CE ϵ C to the true Pareto front is better than those obtained by NSGA-II and MOPSDE. Moreover, the efficient solutions of CE ϵ C have good coverage of the true Pareto front according to the extent of spread achieved between the elements of the Pareto optimal front obtained.

For the Srinivas problem, the results obtained by CE ϵ C are less good in terms of convergence to the true Pareto optimal solutions and in terms of the diversity of the solutions (Δ). On the other side, the CE ϵ C remains competitive compared to the MOPSDE in terms of convergence.

For the ZDT3 problem, the CE ϵ C gets stuck in some discontinuous areas of the objective space. Consequently, the results obtained by CE ϵ C are less good in terms of convergence and diversity of the solutions than the results obtained by NSGA-II and MOPSDE.

5. Conclusion

This paper introduced a new hybrid approach that combines two competitive methods in the optimization area, the stochastic methods, and the deterministic methods. Moreover, the results of our resulting algorithm (CE ϵ C) provide an idea of the validity and the competitiveness of our approach. They also offer a perspective idea to our future and current work.

Acknowledgments

The authors are grateful to the anonymous reviewers for their helpful and valuable comments when reviewing our manuscript. we would like also to thank the organizers of the International Conference on Mathematics and Data Science (ICMDS 2020).

References

- [1] P. Xidonas, C. Hassapis, G. Mavrotas, C. Staikouras and C. Zopounidis, Flots d'Anosov a distributions stable et instable, *Annals of Operations Research*, 267 (2018), 585-606.
- [2] H. Ren, Y. Lu, Q. Wu, X. Yang and A. Zhou, Multi-objective optimization of a hybrid distributed energy system using NSGA-II algorithm, *Frontiers in Energy*, 12 (2018), 518-528.
- [3] J. Huang, Z. Zhou, Z. Gao, M. Zhang and L. Yu, Aerodynamic multi-objective integrated optimization based on principal component analysis, *Chinese Journal of Aeronautics*, 30 (2017), 1336-1348.
- [4] J. Branke, K. Deb, K. Miettinen and S. Roman, *Multiobjective Optimization, Interactive and Evolutionary Approaches*, 1st edition. Springer-Verlag Berlin Heidelberg, 2008.
- [5] A. Lpez Jaimes, S. Zapotecas-Martnez and C. Coello, An Introduction to Multiobjective Optimization Techniques, in *Optimization in Polymer Processing* (eds. A. Gaspar-Cunha, J. A. Covas), Nova Science Publishers, (2011), 29-57.
- [6] Y. Collette and P. Siarry, *Multiobjective Optimization. Principles and Case Studies*, 1st edition, Springer-Verlag Berlin Heidelberg, 2004.
- [7] F. Logist, B. Houska, M. Diehl and J.F. Van Impe, A Toolkit for Multi-Objective Optimal Control in Bioprocess Engineering, *IFAC Proceedings Volumes*, 43 (2010), 281-286.
- [8] Xin Zhang and Xiu Zhang, Cross Entropy Method Meets Local Search for Continuous Optimization Problems, *International Journal on Artificial Intelligence Tools*, 26 (2017).
- [9] Z. Botev, D. Kroese, R. Rubinstein and P. Lécuyer, The Cross-Entropy Method for Optimizations, in *Handbook of Statistics* (eds. Govindaraju. D.Venu and C.R. Rao), Elsevier, (2013), 35-59.
- [10] R.Y. Rubinstein, Optimization of computer simulation models with rare events, *European Journal of Operational Research*, 99 (1997), 89-112.
- [11] R.Y. Rubinstein and D.P. Kroese, *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning*, 1st edition, Springer-Verlag New York, 2004.
- [12] K. Sebaa, A. Tlemani, M. Bouhedda and N. Henini, Multiobjective Optimization Using Cross-Entropy Approach, *Journal of Optimization*, (2013), 1-9.
- [13] I. Plik and T. Terlaky, Interior Point Methods for Nonlinear Optimization, in *Nonlinear Optimization* (eds. I. M. Bomze, V. F. Demyanov, R. Fletcher and T. Terlaky), Springer, Berlin, Heidelberg, (2010), 215-276.
- [14] M. Lalee, J. Nocedal and T. Planteng, On the Implementation of an Algorithm for Large-Scale Equality Constrained Optimization, *SIAM Journal on Optimization*, 8 (1998), 682-706.
- [15] R. H. Byrd, J. C. Gilbert and J. Nocedal, A trust region method based on interior point techniques for nonlinear programming, *Mathematical Programming*, 89 (2000), 149-185.
- [16] Y. Pei and D. Zhu, Local convergence of a trust-region algorithm with line search filter technique for nonlinear constrained optimization, *Applied Mathematics and Computation*, 273 (2016), 797-808.
- [17] T. Binh and U. Korn, A parallel multiobjective evolutionary algorithm, Technical report. Barleben, Germany: Institute for Automation and Communication, 1999.
- [18] M. Tanaka, GA-based decision support system for multi-criteria optimization, *IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, 1995.
- [19] H. Tamaki, H. Kita and S. Kobayashi, Multi-objective optimization by genetic algorithms: a review, *Proceedings of IEEE International Conference on Evolutionary Computation*, 1996.
- [20] J. Nocedal and S.J. Wright, *Numerical Optimization*, 2nd edition, Springer, New York, 2006.
- [21] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, A Fast and Elitist Multiobjective Genetic Algorithm : NSGA-II, *IEEE Transactions on Evolutionary Computation*, 6 (2002), 182-197.
- [22] Y.X. Su and R. Chi, Multi-objective particle swarm-differential evolution algorithm, *Neural Computing and Applications*, 28 (2015), 407-418.
- [23] J. Fliege and A. I.F. Vaz, A method for constrained multiobjective optimization based on SQP techniques, *SIAM Journal on Optimization*, 26 (2019), 2091-2119.
- [24] N. Srinivas and K. Deb, Muultiobjective optimization using nondominated sorting in genetic algorithms, *Evolutionary Computation*, 2 (1994), 221-248.

- [25] E. Zitzler, K. Deb, and L. Thiele, Comparison of Multiobjective Evolutionary Algorithms, *Evolutionary Computation*, 8 (2000), 173–195.
- [26] G. P. Rangaiah and A. Bonilla-Petriciolet, *Multi-objective Optimization in Chemical Engineering: Developments and Applications*, 1st edition, John Wiley & Sons, Ltd, 2013.
- [27] L. Sun, G. DePuy and G. W. Evans, Multi-objective optimization models for patient allocation during a pandemic influenza outbreak, *Computers & Operations Research*, 51 (2014), 350-359.
- [28] Y. Jin, *Multi-Objective Machine Learning*, 1st edition, Springer-Verlag Berlin Heidelberg, 2006.