**A Preference-based Coevolutionary Multiobjective Evolutionary Algorithm to Solve Bilevel Constrained Problems**

**Abstract:** This chapter presents an experimental study on incorporating Decision Maker Preferences into a Coevolutionary version of the NSGAII algorithm, to direct the algorithm towards preferred parts of the Pareto Fronts. ***Brief description of the algorithm.*** For the experiments, Real-World Multiobjective CEC-2021 Constrained Optimization Problems are solved using the proposal and 2 state-of-the-art algorithms. ***The results suggest that..***

**Keywords: Decision Making, Multiobjective Evolutionary Algorithms, Coevolution, Bilevel, Constraints-Handling.**

**Introduction**

In the real world, when it's necessary to optimize or people want to do it to improve certain objects, process or computational modeling problems, questions arise such as: How can I solve the problem? What tools are there to do it? Which one should be used? before thinking further about the results and implementation.

Due to the different properties’ problems have (objectives, unconstrained, constrained, static, dynamic, differentiable, etc.), they can be classified in different ways. If we take as a starting point the number of objectives to be improved, two

kinds of problems can be devised: Single Objective Optimization Problems (SOP's), where a single objective function is optimized (minimize or maximize); and Multiobjective Optimization Problems (MOP's), where it is required to optimize a set of objective functions simultaneously, where some can be maximized and others minimized, and which must be strictly in conflict with each other. Therefore, the notion of Optimality is another difference when solving these problems, which in other words, refers to how determine when one solution is better than another. When solving SOP's, the notion coined is called global optimality, where for the solutions it is easy to notice their superiority between them since they can be global or local optimal. In contrast, when MOPs are solved, the notion of Optimality changes because it is difficult to determine when one solution is better than another due to conflicts between objectives, consequently there must be a compromise relationship between the solutions because a solution cannot improve one without degrading the performance of the others. This notion is known as Pareto Optimality. Inherently, there are different concepts immersed, like as a Pareto dominance, where a solution  dominates iff for all the objective functions, and there must exist at least one function such that . When a solution is not dominate among the rest, it is called as non-dominated. In consequence, a set of non-dominated solutions is called Pareto Optimal Set (PS) and the image they form in the objective functions space is called Pareto Optimal Front (POF). Up to this point, it has been talked about in terms of modeling and model improvement. One part that has been dealt with less frequently is the incorporation of preferences. Because a set of non-dominated solutions is obtained, it is possible that not all of them satisfy what the decision maker requires and is only interested in some of them.

Two areas that provide optimization techniques are: Operations research (or mathematical programing or Multi-criterion Decision Making), where are the classical optimization techniques and Evolutionary Computing, where algorithms inspired by biological evolution are presented.

For example, in Mathematical Programming, it can be classified based on the moment in which the decision maker intervenes to determine if a solution is adequate: (1) A priori methods (goal programming, lexicographic method), (2) a posteriori methods (weighted sums methods, -constraint method) and (3) progressive articulation of preferences (Tchebycheff method, GUESS method) [9]. When a problem is solved and the decision maker wants to intervene, The POF is returned to the decision maker so that he can decide if it meets their needs or not, or if prefer some objective functions more than another and thus direct the search for solutions towards your preference. Evolutionary Computing (EC) is a specialized area that provides optimization algorithms based on biological evolution. To solve MOP's, it provides techniques called Multiobjective evolutionary algorithms (MOEA's) that can be classified into three large groups: 1) Pareto-based, where they use mechanisms to determine, select and rank solutions based on Pareto dominance for example Non-dominated Sorting Genetic Algorithm II (NSGA-II) [10], Strength Pareto Evolutionary Algorithm 2 (SPEA-2) [11] and Pareto Archive Evolutionary Strategy (PAES) [12], 2) Decomposition-based convert MOP's into SOPs to solve them collaboratively, Decomposition-based Multi-Objective Evolutionary Algorithm (MOEA/D) [14] Reference Vector Guided Evolutionary Algorithm (RVEA) [14] and Non-dominated Sorting Genetic Algorithm III (NSGA-III) [15] are some methods belonging to this category and 3) Indicators-based, include performance metrics belonging to multi-objective evolutionary optimization as selection mechanisms.

Mencionar articulos relacionados a la toma de decisiones (una revision del estado del arte)

The main differences between the techniques proposed in each area are summarized in the Table 1.

Table 1: Differences between classical methods and evolutionary algorithms

|  |  |
| --- | --- |
| Classical methods | Evolutionary algorithms |
| Require knowledge of the problem | Require few knowledge of the problem |
| Sensitive to the Pareto Front Shape | They are not sensitive to the shape of the Pareto front |
| A solution is found by run | Obtain multiple elements of the PS in one run |
| It is the easiest way to solve a MOP | Easy to implement |
|  | Preserve non-dominated solutions by generation |
|  | Some methods incorporate elitist mechanisms |

The main objective when solving a MOP is to find a very good approximation to the true POF, and the non-dominated solutions can be well distributed along the front [8,9]. Real-world problems, such as engineering problems, often include constraints. Therefore, in addition to the solutions' optimality, the feasibility of them must be considered (solutions must satisfy all the constraints of the problem).

In the literature, there are studies about different approaches of Constraint-handling techniques. Examples of these are the penalty functions that consist of punishing those solutions that are not in the feasible area, repair algorithms where they take an unfeasible solution and generate a feasible one based on it, and hybrid methods, techniques that combine several mechanisms to handle constraints

Co-evolutionary algorithms take as a source of inspiration the coexistence of populations of individuals, where in some cases, where in some cases

In [1] proposed a two-archive multiobjective evolutionary algorithm, which is used to solve constrained-MOPs. The first archive, called CA, is used to consider the feasibility, in contrast to the second, called DA, whis is used to provide information of the problem in terms of diversity and convergence.

In [2] the authors proposed a general framework to convert a Multiobjective Evolutionary Algorithm into one based on coevolution called CCMO and thus be able to solve Multiobjective problems with the presence of constraints.

The authors proposed the concept of weak coevolution , in which both populations are evolved in isolation, in contrast to most of the coevolutionary algorithms proposed, considering shared information in each step. Como paso inicial, ya se debe contar con el algoritmo a reestructurar y conocer sus componentes. El nuevo enfoque es creado a como sigue: First, two populations are generated, one used to solve the original problem, and the second one to solve a derived problem, the problems without constraints, to allow the algorithm to solve the problem without biasing the search. Then, we proceed to select half of the individuals from each population using the original selection operator of the algorithm. From the selected individuals, using the crossover and mutation operator, the offspring are created. It is important to note that two sets of solutions with different characteristics are obtained, some considering feasibility and others not. Both sets are united from that union, again they are united with each of the parent subpopulations.

***Multiobjective-Optimization Theory***

A Multiobjective Optimization Problem (MOP) is statement as:

Minimize / Maximize:

Subject to:

where is a design vector decision vector or simply called a solution, is the design variables space, is the objective vector composed of objective functions, represents the objective functions space, are the inequality constraints, are the equality constraints and and are the lower and upper bounds that represent the Boundary Constraints.

Definition 1.1 Pareto Dominance: Let be and two decision vectors, if and at least , then it is said dominates .

Definition 1.2 Pareto Optimal Solution: Let be the feasible zone, established in terms of the constrained problem, , if , is a Pareto Optimal Solutions of the problem.

Definition 1.3 Pareto Optimal Set: It is the set composed of all the Pareto optimal Solutions:

Definition 1.4 Pareto Optimal Front: It is the image of in the objective space.

***Assessing Performance***

Many years ago, several research projects were inspired by the creation of metrics to evaluate the approximations' quality (Pareto Fronts) obtained by a MOEA when solving a problem. Various metrics emerged, which can be grouped into two classes: unary and binary indicators. Unary indicators use only one set of solutions that are evaluated based on a certain measured characteristic, for example distance to the True Pareto Front, and assign a numerical value to the set, reflecting its quality in this. Conversely, Binary indicators evaluate a pair of approximations and assign a numerical value to it. Detailed studies about general properties of quality indicators are found in [16,17].

These indicators are functions that evaluate the approximations in a specific characteristic that in the end is represented in a real value. Because there are different characteristics that approximations can have, such as capacity, convergence, diversity, distribution, dispersion, there is a vast set of them [7]. In this work, the indicators selected to evaluate the approximations are hypervolume, inverted generational distance, coverage and R metric. Each of them is briefly described below.

Hypervolume: The hypervolume indicator, also called the Lebesgue measure or S metric, is the only indicator that has the property of being Pareto compliance. The HV quantifies the space dominated by an approximation with respect to a reference point, which is defined as:

where L is the Lebesgue measure, NS is the set of non-dominated solutions, a is a point that belongs to this and R is a reference point.

Inverted-generational distance: IGD measures the quality based on convergence and diversity of an Reference Pareto Front to an approximation obtained by an algorithm. IGD evaluates the average distance from each reference point.

Covering: Binary indicator that quantifies the portion of solutions in A that dominates the solutions in B. Therefore, to compare two fronts, the indicator has to be calculated in both directions.

R metric:

***Multiobjective Evolutionary Algorithms***

Write on different MOEAs approaches (based on dominance, performance indicators, etc).

As mentioned above, Multiobjective Evolutionary Algorithms can be classified into three big groups: algorithms based on pareto dominance, based on decomposition, and based on quality indicators. When a problem has constraints, it is necessary to incorporate a Constraint Handling Techniques so that the algorithm can produce solutions to the problem. Next, a review of algorithms in these aspects is made with the use of constraint-handling techniques.

***Pareto-based MOEA's***

A lot of research has been dedicated on investigating the influence of Constraint-Handling Techniques (CHT's) with Pareto-based MOEA's. Ma et al. [Ma2019] proposed a Constraint-Handling Technique called ToR that calculates the fitness function for each solution as the weighted sum of two rankings: One based on the feasibility of solutions and the other, ranking the solutions based on the dominance principle. In addition, in the weighted sum, they considered to include the proportion of feasible solutions in the current population allowing to trade-off between constraints and objectives during the evolutionary process. They argued this technique allows to quickly enter to the feasible region causing poor diversity like the population could be stuck in local area of the feasible region; and if the feasible region is composed of disjoint regions, the population could lose some of them, causing the loss of optimal Pareto solutions. The authors proposed to include information of the objectives to solve this disadvantage using constants values that represent the percentage of feasible and infeasible solutions in the populations.

Zhang et al. [Zhang2019] propose a CHT called adaptive -truncation that divides the population in two parts, with the intention of allowing non-feasible solutions with a good fitness value participate in the evolutionary process favoring diversity of the population. The proposed technique is added to the NSGA-II algorithm, where in addition present an improvement to the density estimator of such algorithm.

In [Wang2020] propose an algorithm that incorporates the concept of relation of

constrained dominance based on angles to allow infeasible solutions of good quality participates in the evolutionary process. They also present a mechanism of density estimation and a mechanism that considers convergence, diversity, and feasibility of solutions simultaneously.

***Decomposition-based MOEA's***

Zhu et al. \cite{Zhu2019} present an algorithm that contemplates two weight vectors, one for control convergence and another for diversity. The solutions associated with the vectors convergence are updated by minimizing the aggregation function, while the associated with the diversity vectors are updated based on the aggregation function and the sum of constrained violation.

In \cite{Ming2019} a dual-grid dual-phase strategy (DDPS) is proposed, where one population maintains feasible solutions, and the other explores the whole search space without considering constraints. Then, both populations share information and pull each other so as to enable the algorithm to search for the optimal feasible region. In addition, they proposed dual-grid push and pull search strategy in the framework of MOEA/D, abbreviated as

MOEA/D-DPPS.

***Indicator-based MOEA's***

In [Liu2019] carry out the first study on the possibility and rationality of combining evolutionary multi-objective algorithms based on Indicators with techniques for managing restrictions together. Also, they propose 9 algorithms, develop a framework that can combine both techniques easily and suggest how to select the adequate Indicator-based multi-objective evolutionary algorithm. Finally, they perform a comparison of the algorithm that best solved the problems with 5 algorithms of the state of the art.

***Preference-based Multiobjective Evolutionary Algorithms***

Reference point-based approaches, reference direction-based approaches, preference region-based approaches, trade-off-based approaches, objective comparison-based approaches, solution comparison-based approaches, outranking-based approaches, knee point-based approaches []

***Constraint-Handling in Multiobjective Evolutionary Algorithms***

The scientific community has proposed CHTs to provide MOEAs with the purpose of solving constrained multi-objective problems, called CMOEA's (C-MOEA’s) respectively. Some categories are the following:

* Constrained Dominance Principle (CDP): These mechanisms use constraint violation sum to guide the search instead of objective functions, which can lead to premature convergence.
* Penalty functions: These techniques depend on the problem and require and penalty factors tuning.
* Stochastic Ranking (SR): It has an user-dependent parameter called Pf that represents the probability of using the objective function for the comparison of two non-feasible solutions regardless of the constraints violation sum. There may be infeasible solutions at the end of the population.
* \epsilon constraint method: It requires careful decrement control of the \epsilon parameter. It presents difficulties in finding the optimal solutions located in the bounds of the feasible region.
* Methods based on multi-objective optimization: These mechanisms transform a constrained problem with m objectives and k constraints into a multi-objective problem with m + k objectives.
* Hybrid methods: Combination of different CHT's.

In the literature, there are studies about different approaches of Constraint-handling techniques. Examples of these are the penalty functions that consist of punishing those solutions that are not in the feasible area, repair algorithms where they take an unfeasible solution and generate a feasible one based on it, and hybrid methods, techniques that combine several mechanisms to handle constraints. In this work, we adopted the Constraint-dominance principle which is a variation if the feasibility rules:

\begin{itemize}

\item Between two feasible individuals, the one with the best value of the objective function is selected.

\item A feasible individual is preferable to an unfeasible one.

\item Between two non-feasible individuals, the one with the lowest sum of constraint violation is selected. A representation of the SVR is the one expressed in Eq. \ref{ec:sumaRestricciones}

\begin{equation}

\label{ec:sumaRestricciones}

\phi\_{x} = \sum\_{j=1}^{m} max(0,g\_{j}(x))

\end{equation}

where $\phi\_{x} $ refers to CVR.\\

\end{itemize}

**Proposal**

**Experiments**

In the specialized literature we can find optimization problems modeled based on real world problems, and, on the other hand, artificial problems that were created to add different levels of difficulty, based on one or more properties of the problems, to the techniques. In this work, we focus on the Real-World Multiobjective Optimization Problem, in particular, the suite proposed to the CEC2021 [], where a set of 50 RWCMOPs are collected from different scientific and engineering fields.

**Discussion**

**Conclusions**

**References**

[1] Li, K., Chen, R., Fu, G., & Yao, X. (2019). Two-Archive Evolutionary Algorithm for Constrained Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, *23*(2), 303–315. <https://doi.org/10.1109/TEVC.2018.2855411>.

[2] Tian, Y., Zhang, T., Xiao, J., Zhang, X., & Jin, Y. (2021). A Coevolutionary Framework for Constrained Multiobjective Optimization Problems. *IEEE Transactions on Evolutionary Computation*, *25*(1), 102–116. <https://doi.org/10.1109/TEVC.2020.3004012>

[3] Filatovas, E., Kurasova, O., Redondo, J. L., & Fernández, J. (2020). A reference point-based evolutionary algorithm for approximating regions of interest in multiobjective problems. *Top*, *28*(2), 402–423. https://doi.org/10.1007/s11750-019-00535-z

[4] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," in IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, pp. 67-82, April 1997, doi: 10.1109/4235.585893.

[5] Kumar, A., Wu, G., Ali, M. Z., Luo, Q., Mallipeddi, R., Suganthan, P. N., & Das, S. (2021). A Benchmark-Suite of Real-World Constrained Multi-Objective Optimization Problems and some Baseline Results. *Swarm and Evolutionary Computation*, *67*(August 2020), 100961. <https://doi.org/10.1016/j.swevo.2021.100961>

[6] Zitzler E., Thiele L., ” Multiobjective Optimization Using Evolutionary Algorithms - A Com- parative Case Study”, LNCS, vol. 1498, pp. 292301. Springer, Heidelberg (1998)

[7] Falcón-cardona, J. G., & Science, C. (2020). *Indicator-based Multi-objective Evolutionary Algorithms :* *53*(2).

[8] Kalyanmoy Deb and Deb Kalyanmoy. 2001. Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, Inc., USA.

[9] Carlos A. Coello, Gary B. Lamont, and David A. Van Veldhuizen.EvolutionaryAlgorithms for Solving Multi-Objective Problems (Genetic and EvolutionaryComputation). Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387332545

[10] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," in IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197, April 2002, doi: 10.1109/4235.996017.

[11] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the strengthPareto evolutionary algorithm.TIK-report, 103, 2001

[12] . Knowles and D. Corne. The pareto archived evolution strategy: a new baselinealgorithm for pareto multiobjective optimisation.InProceedings of the 1999Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), volume 1,pages 98–105 Vol. 1, 1999.

[13] Q. Zhang and H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based onDecomposition.IEEE Transactions on Evolutionary Computation, 11(6):712–731,2007

[14] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff.A Reference Vector GuidedEvolutionary Algorithm for Many-Objective Optimization.IEEE Transactions onEvolutionary Computation, 20(5):773–791, 2016

[15] K. Deb and H. Jain. An Evolutionary Many-Objective Optimization AlgorithmUsing Reference-Point-Based Nondominated Sorting Approach, Part I: SolvingProblems With Box Constraints.IEEE Transactions on Evolutionary Computation,18(4):577–601, 2014

[16] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M. Fonseca, and Viviane Grunert da Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. IEEE Transactions on Evolutionary Computation, 7(2):117–132, April 2003

[17] Li