

A Survey of Constraint-Handling Techniques Based on Evolutionary Multiobjective Optimization

Efrén Mezura-Montes* and Carlos A. Coello Coello†

* National Laboratory of Advanced Informatics (LANIA A.C.)
Rébsamen No. 80 Col. Centro, Xalapa, Veracruz 91000, MEXICO
emezura@lania.mx

† Evolutionary Computation Group (EVOCINV) at CINVESTAV-IPN
Computer Science Department, Av. IPN No. 2508
Col. San Pedro Zacatenco México D.F. 07300, MÉXICO
ccoello@cs.cinvestav.mx

Abstract. In this paper, we present several constraint-handling techniques based on evolutionary multiobjective optimization concepts. Some basic definitions are presented as well as the way in which a global nonlinear optimization problem is transformed into an unconstrained multiobjective optimization problem. A taxonomy of methods is proposed and each one is described. Some interesting findings regarding common features of such approaches are also discussed.

1 Introduction

Nowadays, evolutionary algorithms (EAs) have become a popular choice to solve different types of optimization problems [1–3]. In fact, this paper points out the application of some ideas originally designed to solve an specific type of optimization problem using EAs which are now applied to solve a different type of problem. Despite being considered powerful search engines, EAs, in their original versions, lack a mechanism to incorporate constraints into the fitness function in order to solve constrained optimization problems. Hence, several approaches have been proposed to deal with this issue. Michalewicz [4] and Coello [5] have presented comprehensive surveys about techniques added to EAs working in constrained search spaces. The most popular method adopted to handle constraints in EAs was taken from the mathematical programming literature: penalty functions (mostly exterior penalty functions). The aim is to decrease (punish) the fitness of infeasible solutions as to favor those feasible individuals in the selection and replacement processes. The main advantage of the use of penalty functions is their simplicity; however, their main shortcoming is that penalty factors, which determine the severity of the punishment, must be set by the user and their values are problem-dependent [6, 5].

This has motivated the design of alternative techniques like those based on special coding and operators [7, 8] and repair algorithms [9]. Unlike penalty functions, which combine the objective function and the constraints values into one fitness value, there are other approaches which handle these two values separately. The most representative approaches, which work based on this idea are: satisfying constraints based on a lexicographic order [10], the superiority of feasible points [11, 12] and the methods based on

evolutionary multiobjective optimization concepts. This paper focuses on the last type of techniques (those based on multiobjective optimization concepts) and describes and criticizes them. The idea of using the Pareto dominance relation to handle constraints was originally suggested by Fonseca and Fleming [13, 14] back in 1995. We propose a classification of these methods, based on the way they transform the nonlinear programming problem (NLP) into a multiobjective optimization problem (MOP):

1. Approaches which transform the NLP into an unconstrained bi-objective optimization problem (the original objective function and the sum of constraint violation).
2. Techniques which transform the NLP into an unconstrained MOP where the original objective function and each constraint of the NLP are treated as separate objectives. From this category, we observed two sub-categories: (1) those methods which use non-Pareto concepts (mainly based on multiple populations) and (2) techniques which use Pareto concepts (ranking and dominance) as selection criteria.

The paper is organized as follows: In Section 2 we present the general NLP, some multiobjective optimization concepts used in this survey and the transformation of the NLP into a MOP. After that, in Section 3 those approaches which solve the problem as a bi-objective problem (using the original objective function and the sum of constraint violation) are presented. Later on, Section 4 shows techniques based on solving the problem by taking the original objective function and each of the constraints of the problem as different objectives, either by using Pareto and non-Pareto concepts. In Section 5, we highlight interesting issues found in our research. Finally, Section 6 presents some conclusions and future paths of research in the area.

2 Problem definition and transformation

In the following definitions we will assume minimization (without loss of generality). The general NLP is defined as to:

$$\text{Find } \mathbf{X} \text{ which minimizes } f(\mathbf{X}) \quad (1)$$

subject to:

$$g_i(\mathbf{X}) \leq 0, \quad i = 1, \dots, m \quad (2)$$

$$h_j(\mathbf{X}) = 0, \quad j = 1, \dots, p \quad (3)$$

where $\mathbf{X} \in \mathbb{R}^n$ is the vector of solutions $\mathbf{X} = [x_1, x_2, \dots, x_n]^T$, where each x_i , $i = 1, \dots, n$ is bounded by lower and upper limits $L_i \leq x_i \leq U_i$ which define the search space \mathcal{S} , \mathcal{F} is the feasible region and $\mathcal{F} \subseteq \mathcal{S}$; m is the number of inequality constraints and p is the number of equality constraints (in both cases, constraints could be linear or nonlinear).

Now, we enumerate some multiobjective optimization concepts used in techniques to handle constraints in EAs to solve the NLP.

The general multiobjective optimization problem (MOP) is defined as to:

$$\text{Find } \mathbf{X} \text{ which minimizes } \mathbf{F}(\mathbf{X}) = [f_1(\mathbf{X}), f_2(\mathbf{X}), \dots, f_k(\mathbf{X})]^T \quad (4)$$

subject to:

$$g_i(\mathbf{X}) \leq 0, \quad i = 1, \dots, m \quad (5)$$

$$h_j(\mathbf{X}) = 0, \quad j = 1, \dots, p \quad (6)$$

where $\mathbf{X} \in \mathbb{R}^n$ is the vector of solutions $\mathbf{X} = [x_1, x_2, \dots, x_n]^T$, where each x_i , $i = 1, \dots, n$ is bounded by lower and upper limits $L_i \leq x_i \leq U_i$ which define the search space \mathcal{S} , \mathcal{F} is the feasible region and $\mathcal{F} \subseteq \mathcal{S}$; m is the number of inequality constraints and p is the number of equality constraints (in both cases, constraints could be linear or nonlinear).

When solving NLPs with EAs, equality constraints are usually transformed into inequality constraints of the form:

$$|h_j(\mathbf{x})| - \epsilon \leq 0 \quad (7)$$

where ϵ is the tolerance allowed (a very small value). In the rest of the paper we will refer only to inequality constraints because we will assume this transformation.

In a multiobjective problem, the optimum solution consists on a set of (“trade-off”) solutions, rather than a single solution as in global optimization. This optimal set is known as the Pareto Optimal set and is defined as follows:

$$\mathcal{P}^* := \{\mathbf{X} \in \mathcal{F} \mid \neg \exists \mathbf{X}' \in \mathcal{F} \mathbf{F}(\mathbf{X}') \preceq \mathbf{F}(\mathbf{X})\} \quad (8)$$

where the Pareto dominance (denoted by \preceq) is defined as follows:

A vector $\mathbf{U} = (u_1, \dots, u_k)$ is said to dominate $\mathbf{V} = (v_1, \dots, v_k)$ (denoted by $\mathbf{U} \preceq \mathbf{V}$) if and only if \mathbf{U} is partially less than \mathbf{V} , i.e., $\forall i \in \{1, \dots, k\}, u_i \leq v_i \wedge \exists i \in \{1, \dots, k\} : u_i < v_i$.

Two different ways to transform the NLP into a MOP have been found in the literature, giving us an option to propose a classification of techniques. The first approach transforms the NLP into an unconstrained bi-objective problem. The first objective is the original objective function and the second one is the sum of constraint violation as follows: optimize $\mathbf{F}(\mathbf{X}) = (f(\mathbf{X}), G(\mathbf{X}))$, where $G(\mathbf{X}) = \sum_{i=1}^{m+p} \max(0, g_i(\mathbf{X}))$. Unlike typical MOPs, when solving a transformed NLP, we are not looking for a set of solutions (as described in Equation 8). Instead, we seek a unique solution, the global constrained optimum, where: $f(\mathbf{X}) \leq f(\mathbf{Y})$ for all feasible \mathbf{Y} and $G(\mathbf{X}) = 0$.

The second approach transforms the problem into an unconstrained MOP, in which we will have $k + 1$ objectives, where k is the total number of constraints ($m + p$) and the additional objective is the original NLP objective function. Then, we can apply multiobjective optimization concepts to the new vector $\mathbf{F}(\mathbf{X}) = (f(\mathbf{X}), g_1(\mathbf{X}), \dots, g_{m+p}(\mathbf{X}))$, where $g_1(\mathbf{X}), \dots, g_{m+p}(\mathbf{X})$ are the original constraints of the problem.

As mentioned before, we are looking for the global constrained optimum instead of a set of trade-off solutions. Then, we require the following: $g_i(\mathbf{X}) = 0$ for $1 \leq i \leq (m + p)$ and $f(\mathbf{X}) \leq f(\mathbf{Y})$ for all feasible \mathbf{Y} .

This change on the ultimate goal of the technique prompts to changes in the way the multiobjective concepts are applied i.e. how nondominance, Pareto ranking and multipopulation-based techniques are used. In the next Sections we will describe those approaches proposing different ways to deal with this issue.

3 Techniques solving a bi-objective optimization problem

Surry & Radcliffe [15] proposed COMOGA (Constrained Optimization by Multiobjective Optimization Genetic Algorithms) where individuals are Pareto-ranked based on the sum of constraint violation. Then, solutions can be selected using binary tournament selection based either on their rank or their objective function value. This decision is based on a parameter called P_{cost} whose value is modified dynamically. The aim of the proposed approach to solve this bi-objective problem is based on reproducing solutions which are good in one of the two objectives with other competitive solutions in the other objective e.g. constraint violation (as Shaffer's VEGA promoted to solve MOPs [16]). COMOGA was tested on a gas network design problem providing slightly better results than those provided by a penalty function approach. Its main drawbacks are that it requires several extra parameters and that it has not been tested extensively.

Camponogara & Talukdar [17] proposed to solve the bi-objective problem in the following way: Based on the Pareto Sets generated, an operator that substitutes crossover takes two Pareto sets S_i and S_j where $i < j$ and two solutions $x_i \in S_i$ and $x_j \in S_j$ where x_i dominates x_j . With these two points a search direction is defined using:

$$d = \frac{(x_i - x_j)}{|x_i - x_j|} \quad (9)$$

A line search begins by projecting d over one variable axis on decision variable space in order to find a new solution x which dominates both x_i and x_j . At pre-defined intervals, the worst half of the population is replaced with new random solutions to avoid premature convergence. This indicates some of the problems of the approach to maintain diversity. Additionally, the use of line search within a GA adds some extra computational cost. Furthermore, it is not clear what is the impact of the segment chosen to search in the overall performance of the algorithm.

Coello [18] proposed a ranking procedure based on a counter which was incremented based on the number of individuals in the population which dominated a given solution based on several criteria (feasibility, sum of constraint violation and number of constraints violated). The approach was tested on a set of engineering design problems providing competitive results. Some adaptive mechanism was implemented to tune their parameters. Its main drawback is the computational cost of the technique and its difficulties to handle equality constraints [19].

Zhou et al. [20] proposed a ranking procedure based on Pareto Strength [21] for the bi-objective problem, i.e. to count the number of individuals which are dominated for a given solution. Ties are solved by the sum of constraint violation (second objective in the problem). The Simplex crossover (SPX) operator is used to generate a set of offspring where the individual with the highest Pareto strength and also the solution with the lowest sum of constraint violation are both selected to take part in the population for the next generation. The approach was tested on a subset of the well-known benchmark for evolutionary constrained optimization [22]. The results were competitive but using different set of parameters for different functions, which made evident the sensitivity of the approach to the values of its parameters.

Wang and Cai [23] used a framework similar to the one proposed by Zhou et al. [20] because they also used the SPX with a set of parents to generate a set of offspring.

However, instead of using just two offspring from the set of offspring, all nondominated solutions (in the bi-objective space) are used to replace the dominated solutions in the parent population. Furthermore, they use an external archive to store infeasible solutions with a low sum of constraint violation in order to replace some random solutions in the current population. The idea is to maintain infeasible solutions close to the boundaries of the feasible region in order to sample this region as to find optimum solutions located there [24]. The approach provided good results in 13 well-known test problems. However, different set of values for the parameters were used, depending of the dimensionality of the problem.

Venkatraman and Yen [25] proposed a generic framework to solve the NLP. Their approach is divided in two phases: The first one treats the NLP as a constraint satisfaction problem i.e. the goal is to find at least one feasible solution. To achieve that, the population is ranked based only on the sum of constraint violation. The second phase starts when the first feasible solution was found. Now both objectives (original objective function and the sum of constraint violation) are taken into account and nondominated sorting [26] is used to rank the population (alternatively, the authors proposed a preference scheme based on feasibility rules [12], but nondominated sorting provided better results). Also, to favor diversity, a niching scheme based on the distance of the nearest neighbors to each solution is applied. To decrease the effect of differences in values, all constraints are normalized before calculating the sum of those which are violated. The approach used a typical GA as a search engine with 10% elitism. The approach provided good quality results in 11 well-known benchmark problems and in some problems generated with the Test-Case Generator tool [27], but lacked consistency due to the fact that the way to approach the feasible region is mostly at random because of the first phase which only focuses on finding a feasible solution, regardless of the region from which the feasible region is approached.

Wang et al. [28] also solved the bi-objective problem but using selection criteria based on feasibility very similar to those proposed by Deb [12], where a feasible solution is preferred over an infeasible one; between two feasible solutions, the one with the best objective function value is selected and finally, between two infeasible solutions, the one with the lowest sum of constraint violation is chosen. Furthermore, they proposed a new crossover operator based on uniform design methods [28]. This operator is able to explore regions closer to the parents. Finally, a Gaussian noise is used as a mutation operator. The approach was tested on a subset of the well-known benchmark used to test evolutionary algorithms in constrained optimization [22]. No details are given in the paper about the influence of the extra parameters required to control the crossover operator (q) and the number of offspring generated (r).

4 Techniques solving a multiobjective problem with objective function and constraints as separate objectives

4.1 Techniques based on non-Pareto schemes

Parmee & Purchase [29] used the idea proposed in VEGA [16] to guide the search of an evolutionary algorithm to the feasible region of an optimal gas turbine design

problem with a heavily constrained search space. The aim of VEGA is to divide the population into sub-populations, and each sub-population will have the goal to optimize one objective. In this case, the set of objectives are only the constraints of the problem. Genetic operators are applied to all solutions regardless of the sub-population of each solution. In Parmee's approach, once the feasible region is reached, special operators are used to improve feasible solutions. The use of these special operators that preserve feasibility make this approach highly specific to one application domain rather than providing a general methodology to handle constraints.

Coello [30] also used VEGA's idea [16] to solve NLPs. At each generation, the population was split into $m + 1$ sub-populations of equal fixed size, where m is the number of constraints of the problem. The additional sub-population handles the objective function of the problem and the individuals contained within it are selected based on the unconstrained objective function value. The m remaining sub-populations take one constraint of the problem each as their fitness function. The aim is that each of the sub-populations tries to reach the feasible region corresponding to one individual constraint. By combining these different sub-populations, the approach will reach the feasible region of the problem considering all of its constraints. The main drawback of the approach is that the number of sub-populations increases linearly with respect to the number of constraints.

This issue was indeed tackled by Liang and Suganthan [31], where a dynamic particle multi-swarm optimization was proposed. They also used VEGA's idea to split the swarm into sub-swarms and each sub-swarm optimized one objective. However, in this case, the sub-swarms are assigned dynamically. In this way, the number of sub-swarms depends on the complexity of the constraints to be satisfied instead of the number of constraints. The authors also included a local search mechanism based on sequential quadratic programming to improve values of a set of randomly chosen *pbest* values. The approach provided competitive results in the extended version of a well-known benchmark adopted for evolutionary constrained optimization [31]. The main drawbacks of the approach are that it requires extra parameters to be tuned by the user and it also presented a strong dependency on the local search mechanism.

4.2 Techniques based on Pareto schemes

Jiménez et al. [32] proposed an approach that transforms the NLP (and also the constraint satisfaction and goal programming problems) into a MOP by assigning priorities. Regarding the NLP, constraints are assigned a higher priority than the objective function. Then, a multiobjective algorithm based on a pre-selection scheme is applied. This algorithm generates from two parents a set of offspring which will be also mutated to generate another set. The best individual from the first set of offspring (non-mutated) and the best one of the mutated ones, will replace each of the two parents. The idea is to favor the generation of individuals close to their parents and to promote implicit niching. Comparisons among individuals are made by using dominance. A real-coded GA was used as a search engine with two types of crossover operators (uniform and arithmetic) and two mutation operators (uniform and non-uniform). The results on 11 problems taken from a well-known benchmark [22] were promising. The main drawback of the approach is the evident lack of knowledge about the effect of the parameter

“ q ” related with the pre-selection scheme and also the number of evaluations performed by the approach in each test problem because such information is not provided.

Ray et al. [33, 34] proposed the use of a Pareto ranking approach that operates on three spaces: objective space, constraint space and the combination of the two previous spaces. This approach also uses mating restrictions to ensure better constraint satisfaction in the offspring generated and a selection process that eliminates weaknesses in any of these spaces. To maintain diversity, a niche mechanism based on Euclidean distances is used. This approach can solve both constrained or unconstrained optimization problems with one or several objective functions. The mating restrictions used by this method are based on the information that each individual has about its own feasibility. Such a scheme is based on an idea proposed by Hinterding and Michalewicz [35]. The main advantage of this approach is that it requires a very low number of fitness function evaluations with respect to other state-of-the-art approaches. Its main drawback is that its implementation is considerably more complex than that of any of the other techniques previously discussed.

Ray extended his work to a simulation of social behavior [36, 37], where a societies-civilization model is proposed. Each society has its leaders which influence their neighbors. Also, the leaders can migrate from one society to another, promoting exploration of new regions of the search space. Constraints are handled by a nondominated sorting mechanism [26] in the constraints space. A leader centric operator is used to generate movements of the neighbors influenced by their leaders. The main drawback of the approach is its high computational cost derived from the nondominated sorting and a clustering technique required to generate the societies. Results reported on some engineering design problems are very competitive. However it has not been compared against state-of-the-art approaches adopting the same benchmark [22].

Coello and Mezura [38] implemented a version of the Niche-Pareto Genetic Algorithm (NPGA) [39] to handle constraints in single-objective optimization problems. The NPGA is a multiobjective optimization approach in which individuals are selected through a tournament based on Pareto dominance. However, unlike the NPGA, Coello and Mezura’s approach does not require niches (or fitness sharing [40]) to maintain diversity in the population. Instead it requires an additional parameter called S_r that controls the diversity of the population. S_r indicates the proportion of parents selected by four comparison criteria (based on Deb proposal [12]), but when both solutions are infeasible, a dominance criterion in the constraints space is used to select the best solution. The remaining $1 - S_r$ parents will be selected by a pure probabilistic approach. Results indicated that the approach was robust, efficient and effective. However, it was also found that the approach had scalability problems (its performance degrades as the number of decision variables increases).

The use of dominance to select between two infeasible solutions was taken to Differential Evolution by Kukkonen and Lampinen [41]. In their approach, when the comparison between parent and offspring is performed and both of them are infeasible, a dominance criterion is applied. The results on the extended version of the benchmark [41] were very competitive.

Angantyr et al. [42] proposed to assign a fitness value to solutions based on a two-ranking mechanism. The first rank is assigned according to the objective function value

(regardless of feasibility). The second rank is assigned by using nondominated sorting [26] in the constraints space. These ranks have adaptive weights when defining the fitness value. The aim is to guide the search to the unconstrained optimum solution if there are many feasible solutions in the current population. If the rate of feasible solutions is low, the search will be biased to the feasible region. The goal is to promote an oscillation of the search between the feasible and infeasible regions of the search space. A typical GA with BLX crossover was used. The main advantage of this approach is that it does not add any extra parameter to the algorithm. However, it presented some problems when solving functions with equality constraints [42].

Hernandez et al. [43] proposed an approach named IS-PAES which is based on the Pareto Archive Evolution Strategy (PAES) originally proposed by Knowles and Corne [44]. IS-PAES uses an external memory to store the best set of solutions found. Furthermore, IS-PAES requires a shrinking mechanism to reduce the search space. The multiobjective concept is used in this case as a secondary criterion (Pareto dominance is used only to decide whether or not a new solution is inserted in the external memory). The authors acknowledge that the most important mechanisms of IS-PAES are its shrinking procedure and the information provided by the external memory which is used to decide the shrinking of the search space. Furthermore, despite its good performance as a global optimizer, IS-PAES is an approach far from simple to implement.

Runarsson and Yao [45] presented a comparison of two versions of Pareto ranking in constraint space: (1) considering the objective function value in the ranking process and (2) without considering it. These versions were compared against a typical over-penalized penalty function approach. The authors found in their work that using Pareto Ranking leads to bias-free search, then, they concluded that it causes the search to spend most of the time searching in the infeasible region; therefore, the approach is unable to find feasible solutions (or finds feasible solutions with a poor value of the objective function).

Oyama et al. [46] used a similar approach than the proposed by Mezura and Coello [38]. However, the authors propose to use a set of criteria based on feasibility to rank all the population (instead of using them in a tournament [38]). Moreover, this approach is designed to solve also constrained multiobjective optimization problems. A real-coded GA with BLX crossover was used as the search engine. This technique was used to solve one engineering design problem and also a real-world NLP. No further experiments or comparisons were provided.

5 Remarks

Based on the features found in each of the methods, we highlight the following findings:

- At least in our research, the number of methods which use a MOP, with the objective function and constraints as separate objectives is higher than the total number of approaches that used the bi-objective problem.
- The use of sub-populations has been the less popular.
- There is certain preference to use mean-centric crossover operators (BLX [42, 46], random-mix [33, 34], SPX [23, 20]) over using parent-centric crossover (uniform design methods [28], leader centric operator [36, 37]) when using real-coded

GAs. Furthermore, other authors used more than one crossover operator (uniform and arithmetic [32]). This choice may contradict the findings about competitive crossover operators when using other constraint-handling techniques as GENOCOP and penalty functions [47, 48].

- The use of diversity mechanisms is found in most approaches [15, 17, 23, 25, 33, 34, 36–38, 42, 43].
- The use of explicit local search mechanisms is still scarce ([31]).
- The difficulty of using Pareto concepts when solving the NLP pointed out by Runarsson and Yao [45] has been confirmed by other researchers like Mezura and Coello [19]. However, the methods described in this survey provide several alternatives to deal with the inherent shortcoming for the lack of bias provided by Pareto ranking.

6 Conclusions

A detailed survey of constraint-handling techniques based on multiobjective optimization concepts has been presented. A classification of techniques depending of the type of transformation made from the NLP to either a bi-objective (objective function and sum of constraint violation) or a MOP (with the objective function and each constraint considered as separate objectives) has been proposed. We have presented a discussion about the main features of each method (selection criteria, diversity handling, genetic operators, advantages and disadvantages, experimentation). Furthermore, some interesting findings about all methods have been summarized and briefly discussed. Based precisely of these issues found, we visualize the following paths of future research in the area: (1) a more intensive use of explicit local search mechanisms, (2) an in-depth study of the influence of the genetic operators used in these types of methods, (3) novel and more effective proposals of diversity mechanisms, (4) the combination of multi-objective concepts (Pareto methods with population-based techniques) in one single approach.

Acknowledgments: The second author acknowledges support from CONACyT through project number 42435-Y.

References

1. Goldberg, D.: The Design of Innovation. Kluwer Academic Publishers, New York (2002) ISBN 1-4020-7098-5.
2. Michalewicz, Z., Fogel, D.B.: How to Solve It: Modern Heuristics. 2nd edn. Springer, Germany (2004)
3. Reeves, C.B.: Modern Heuristic Techniques for Combinatorial Problems. John Wiley & Sons, Great Britain (1993)
4. Michalewicz, Z., Schoenauer, M.: Evolutionary Algorithms for Constrained Parameter Optimization Problems. *Evolutionary Computation* **4**(1) (1996) 1–32
5. Coello, C.A.C.: Theoretical and Numerical Constraint Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art. *Computer Methods in Applied Mechanics and Engineering* **191**(11-12) (2002) 1245–1287

6. Smith, A.E., Coit, D.W.: Constraint Handling Techniques—Penalty Functions. In Bäck, T., Fogel, D.B., Michalewicz, Z., eds.: *Handbook of Evolutionary Computation*. Oxford University Press and Institute of Physics Publishing (1997)
7. Michalewicz, Z., Janikow, C.Z.: Handling Constraints in Genetic Algorithms. In Belew, R.K., Booker, L.B., eds.: *Proceedings of the Fourth International Conference on Genetic Algorithms (ICGA-91)*, San Mateo, California, University of California, San Diego, Morgan Kaufmann Publishers (1991) 151–157
8. Schoenauer, M., Michalewicz, Z.: Evolutionary Computation at the Edge of Feasibility. In Voigt, H.M., Ebeling, W., Rechenberg, I., Schwefel, H.P., eds.: *Proceedings of the Fourth Conference on Parallel Problem Solving from Nature (PPSN IV)*, Heidelberg, Germany, Berlin, Germany, Springer-Verlag (1996) 245–254
9. Michalewicz, Z., Nazhiyath, G.: Genocop III: A co-evolutionary algorithm for numerical optimization with nonlinear constraints. In Fogel, D.B., ed.: *Proceedings of the Second IEEE International Conference on Evolutionary Computation*, Piscataway, New Jersey, IEEE Press (1995) 647–651
10. Schoenauer, M., Xanthakis, S.: Constrained GA Optimization. In Forrest, S., ed.: *Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA-93)*, San Mateo, California, University of Illinois at Urbana-Champaign, Morgan Kaufmann Publishers (1993) 573–580
11. Powell, D., Skolnick, M.M.: Using genetic algorithms in engineering design optimization with non-linear constraints. In Forrest, S., ed.: *Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA-93)*, San Mateo, California, University of Illinois at Urbana-Champaign, Morgan Kaufmann Publishers (1993) 424–431
12. Deb, K.: An Efficient Constraint Handling Method for Genetic Algorithms. *Computer Methods in Applied Mechanics and Engineering* **186**(2/4) (2000) 311–338
13. de Fonseca, C.M.M.: Multiobjective Genetic Algorithms with Applications to Control Engineering Problems. PhD thesis, Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK (1995)
14. Fonseca, C.M., Fleming, P.J.: Multiobjective optimization and multiple constraint handling with evolutionary algorithms—Part I: A unified formulation. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans* **28**(1) (1998) 26–37
15. Surry, P.D., Radcliffe, N.J.: The COMOGA Method: Constrained Optimisation by Multiobjective Genetic Algorithms. *Control and Cybernetics* **26**(3) (1997) 391–412
16. Schaffer, J.D.: Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In: *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, Lawrence Erlbaum (1985) 93–100
17. Camponogara, E., Talukdar, S.N.: A Genetic Algorithm for Constrained and Multiobjective Optimization. In Alander, J.T., ed.: *3rd Nordic Workshop on Genetic Algorithms and Their Applications (3NWGA)*, Vaasa, Finland, University of Vaasa (1997) 49–62
18. Coello Coello, C.A.: Constraint-handling using an evolutionary multiobjective optimization technique. *Civil Engineering and Environmental Systems* **17** (2000) 319–346
19. Mezura-Montes, E., Coello, C.A.C.: Multiobjective-Based Concepts to Handle Constraints in Evolutionary Algorithms. In Chávez, E., Favela, J., Mejía, M., Oliart, A., eds.: *Proceedings of the Fourth Mexican International Conference on Computer Science (ENC'2003)*, Los Alamitos, CA, Apizaco, Tlaxcala, México, IEEE Computer Society (2003) 192–199
20. Zhou, Y., Li, Y., He, J., Kang, L.: Multi-objective and MGG Evolutionary Algorithm for Constrained Optimization. In: *Proceedings of the Congress on Evolutionary Computation 2003 (CEC'2003)*. Volume 1., Piscataway, New Jersey, Canberra, Australia, IEEE Service Center (2003) 1–5

21. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation* **3**(4) (1999) 257–271
22. Mezura-Montes, E., Coello Coello, C.A.: A Simple Multimembered Evolution Strategy to Solve Constrained Optimization Problems. *IEEE Transactions on Evolutionary Computation* **9**(1) (2005) 1–17
23. Yong, W., Zixing, C.: A Constrained Optimization Evolutionary Algorithm Based on Multi-objective Optimization Techniques. In: 2005 IEEE Congress on Evolutionary Computation (CEC'2005). Volume 2., Edinburgh, Scotland, IEEE Service Center (2005) 1081–1087
24. Mezura-Montes, E., Coello Coello, C.A.: Adding a Diversity Mechanism to a Simple Evolution Strategy to Solve Constrained Optimization Problems. In: Proceedings of the Congress on Evolutionary Computation 2003 (CEC'2003). Volume 1., Piscataway, New Jersey, Canberra, Australia, IEEE Service Center (2003) 6–13
25. Venkatraman, S., Yen, G.G.: A Generic Framework for Constrained Optimization Using Genetic Algorithms. *IEEE Transactions on Evolutionary Computation* **9**(4) (2005)
26. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**(2) (2002) 182–197
27. Michalewicz, Z., Deb, K., Schmidt, M., Stidsen, T.: Test-Case Generator for Nonlinear Continuous Parameter Optimization Techniques. *IEEE Transactions on Evolutionary Computation* **4**(3) (2000) 197–215
28. Wang, Y., Liu, D., Cheung, Y.M.: Preference bi-objective evolutionary algorithm for constrained optimization. In et al., Y.H., ed.: Computational Intelligence and Security. International Conference, CIS 2005. Volume 3801., Xi'an, China, Springer-Verlag (2005) 184–191 *Lecture Notes in Artificial Intelligence*.
29. Parmee, I.C., Purchase, G.: The development of a directed genetic search technique for heavily constrained design spaces. In Parmee, I.C., ed.: *Adaptive Computing in Engineering Design and Control-'94*, Plymouth, UK, University of Plymouth (1994) 97–102
30. Coello Coello, C.A.: Treating Constraints as Objectives for Single-Objective Evolutionary Optimization. *Engineering Optimization* **32**(3) (2000) 275–308
31. Liang, J., Suganthan, P.: Dynamic Multi-Swarm Particle Swarm Optimizer with a Novel Constraint-Handling Mechanism. In: 2006 IEEE Congress on Evolutionary Computation (CEC'2006), Vancouver, Canada, IEEE Press (2006) 316–323
32. Jiménez, F., Gómez-Skarmeta, A., Sánchez, G.: How Evolutionary Multi-objective Optimization can be used for Goals and Priorities based Optimization. In: *Primer Congreso Español de Algoritmos Evolutivos y Bioinspirados (AEB'02)*, Mérida España (2002) 460–465
33. Ray, T., Kang, T., Chye, S.K.: An Evolutionary Algorithm for Constrained Optimization. In Whitley, D., Goldberg, D., Cantú-Paz, E., Spector, L., Parmee, I., Beyer, H.G., eds.: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2000)*, San Francisco, California, Morgan Kaufmann (2000) 771–777
34. Ray, T., Tai, K.: An Evolutionary Algorithm with a Multilevel Pairing Strategy for Single and Multiobjective Optimization. *Foundations of Computing and Decision Sciences* **26** (2001) 75–98
35. Hinterding, R., Michalewicz, Z.: Your Brains and My Beauty: Parent Matching for Constrained Optimisation. In: *Proceedings of the 5th International Conference on Evolutionary Computation*, Anchorage, Alaska (1998) 810–815
36. Akhtar, S., Tai, K., Ray, T.: A Socio-Behavioural Simulation Model for Engineering Design Optimization. *Engineering Optimization* **34**(4) (2002) 341–354

37. Ray, T., Liew, K.: Society and Civilization: An Optimization Algorithm Based on the Simulation of Social Behavior. *IEEE Transactions on Evolutionary Computation* **7**(4) (2003) 386–396
38. Coello, C.A.C., Mezura-Montes, E.: Handling Constraints in Genetic Algorithms Using Dominance-Based Tournaments. In Parmee, I., ed.: *Proceedings of the Fifth International Conference on Adaptive Computing in Design and Manufacture (ACDM'2002)*. Volume 5., University of Exeter, Devon, UK, Springer-Verlag (2002) 273–284
39. Horn, J., Nafpliotis, N., Goldberg, D.E.: A Niche Pareto Genetic Algorithm for Multiobjective Optimization. In: *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*. Volume 1., Piscataway, New Jersey, IEEE Service Center (1994) 82–87
40. Deb, K.: Genetic algorithms in multimodal function optimization. Master's thesis, University of Alabama, Alabama, USA (1989)
41. Kukkonen, S., Lampinen, J.: Constrained Real-Parameter Optimization with Generalized Differential Evolution. In: *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, Vancouver, Canada, IEEE Press (2006) 911–918
42. Angantyr, A., Andersson, J., Aidanpaa, J.O.: Constrained Optimization based on a Multiobjective Evolutionary Algorithms. In: *Proceedings of the Congress on Evolutionary Computation 2003 (CEC'2003)*. Volume 3., Piscataway, New Jersey, Canberra, Australia, IEEE Service Center (2003) 1560–1567
43. Hernández-Aguirre, A., Botello-Rionda, S., Coello Coello, C.A., Lizárraga-Lizárraga, G., Mezura-Montes, E.: Handling Constraints using Multiobjective Optimization Concepts. *International Journal for Numerical Methods in Engineering* **59**(15) (2004) 1989–2017
44. Knowles, J.D., Corne, D.W.: Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation* **8**(2) (2000) 149–172
45. Runarsson, T.P., Yao, X.: Search biases in constrained evolutionary optimization. *IEEE Transactions on Systems, Man, and Cybernetics Part C—Applications and Reviews* **35**(2) (2005) 233–243
46. Oyama, A., Shimoyama, K., Fujii, K.: New Constraint-Handling Method for Multi-Objective Multi-Constraint Evolutionary Optimization and Its Application to Space Plane Design. In Schilling, R., Haase, W., Periaux, J., Baier, H., Bugada, G., eds.: *Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems (EUROGEN 2005)*, Munich, Germany (2005)
47. Ortiz-Boyer, D., Hervás-Martínez, C., García-Pedrajas, N.: Crossover Operator Effect in Functions Optimization with Constraints. In Merelo-Guervós, J.J., et al., eds.: *Proceedings of the 7th Parallel Problem Solving from Nature (PPSN VII)*, Heidelberg, Germany, Granada, Spain, Springer-Verlag (2002) 184–193 *Lecture Notes in Computer Science* Vol. 2439.
48. Ortiz-Boyer, D., Del-Castillo-Gomariz, R., Garcia-Pedrajas, N., Hervas-Martinez, C.: Crossover effect over penalty methods in function optimization with constraints. In: *2005 IEEE Congress on Evolutionary Computation (CEC'2005)*. Volume 2., Edinburgh, Scotland, IEEE Service Center (2005) 1127–1134