Solving Constrained Multi-objective Optimization Problems with Evolutionary Algorithms

Frikkie Snyman and Mardé Helbig(✉)

University of Pretoria, Pretoria, South Africa
u13028741@tuks.co.za, mhelbig@cs.up.ac.za

Abstract. Most optimization problems in real-life have multiple constraints. Constrained optimization problems with more than one objective, with at least two objectives in conflict with one another, are referred to as constrained multi-objective optimization problems (CMOPs). Two main approaches to solve constrained problems are to add a penalty to each objective function and then optimizing the new adapted objective function, or to adapt the Pareto-dominance principle that are used to compare two solutions in such a way that constraint violations are taken into consideration. This paper investigates how these two approaches affect the performance of the steady-state non-dominated sorting genetic algorithm II (SNSGAII), the Pareto-archived evolution strategy (PAES), the multi-objective evolutionary algorithm based on decomposition (MOEA/D) and a cultural algorithm (CA) when solving CMOPs. The results indicate that there is no statistical significant difference in performance between these two approaches. However, depending on the multi-objective evolutionary algorithm (MOEA) one approach does provide slightly better solutions than the other approach.

1 Introduction

Many optimization problems have more than one objective, with at least two objectives in conflict with one another. Therefore, improving on one objective leads to a weakening of another objective. This kind of problems are referred to as multi-objective optimization problems (MOPs). Evolutionary algorithms (EAs) are computational intelligence algorithms that are based on the natural model of evolution and are commonly used to solve MOPs [3]. Constrained multi-objective optimization problems CMOP refer to MOPs with inequality and/or equality constraints, in addition to boundary constraints of unconstrained MOPs.

Constrained MOPs are more difficult to solve, since the constraints limit the number of valid solutions in the search space, i.e. the feasible space. Two approaches that are frequently used to solve CMOPs with a (MOEA) are:

- Adding a penalty term (function) to each objective function and then optimizing the adapted objective functions [21]

Y. Tan et al. (Eds.): ICSI 2017, Part II, LNCS 10386, pp. 57–66, 2017.
DOI: 10.1007/978-3-319-61833-3_7

- Adapting the Pareto-dominance [8,16] equations to take into account the violation of constraints [5]

The problem with the penalty appraoch is that you are changing the objective function by adding an additional term, the penalty term, to the objective function. Since the various objective functions of the CMOP can have different orders of magnitude, the order of magnitude of the penalty term should also differ. However, the value selected for the penalty term will also influence the quality of the solutions that are found, since the penalty term balances the influence of constraint violation on the fitness of a specific solution. A too small contribution by the penalty term to the fitness value may lead to a set of infeasible solutions being found. In contrast, a too high penalty term may lead to a poor distribution of found solutions [5].

This paper investigates the influence of these two constraint-handling approaches on the steady-state non-dominated sorting genetic algorithm II (SNSGAII), the Pareto-archived evolution strategy (PAES), the multi-objective evolutionary algorithm based on decomposition (MOEA/D) and the cultural algorithm (CA) when solving CMOPs with various characteristics.

The rest of the paper's layout is as follows: Sect. 2 provides an overview of CMOP, the algorithms and the constraint-handling approaches. The experimental setup is discussed in Sect. 3. Section 4 presents the results of the study and conclusions are highlighted in Sect. 5.

2 Background

This section discusses CMOP, as well as the algorithms and constraint-handling approaches that were used in the study.

2.1 Constrained Multi-objective Optimization

A CMOP can be defined as follows:

$$\begin{aligned} \mathit{minimize}: & \, f_k(\mathbf{x}),\ k = 1, 2, \ldots, n_k \\ \mathit{subject\ to}: & \, g_m(\mathbf{x}) \geq 0,\ m = 1, \ldots, n_g \\ & h_m(\mathbf{x}) = 0,\ m = n_g + 1, \ldots, n_g + n_h \\ & x_i^L \leq x_i \geq x_i^U,\ i = 1, 2, \ldots, n_x \end{aligned} \tag{1}$$

where f_k represents the k-th objective function, and g_m and h_m represent the m-th inequality and equality constraint respectively. The i-th component of the upper boundary value of $\mathbf{x}$ is indicated by $\mathbf{x}_i^U$ and the lower boundary value by $\mathbf{x}_i^L$.

Due to the inequality and equality constraints in Eq. 1 not all solutions in the search space will be feasible. Various approaches exist to deal with CMOP [5, 11, 18] and to guide the search of the algorithm towards feasible solutions. Two approaches that are simple to implement are the following:

- Adding a penalty term to the objective function
- Modifying the Pareto-dominance equations to consider constraint violations

Assuming that you minimize all objective functions (a maximizing objective function can be managed by converting it to a minimization function using the duality principle), all constraints are first normalized before the constraint violation is calculated. Let the normalized constraint functions be represented by $g^*_m(\mathbf{x}_i) \geq 0$ for $m = 1, 2, \ldots, n_g$. Then for each solution $\mathbf{x}_i$ the constraint violation is calculated as follows [5]:

$$\omega_m(\mathbf{x}_i) = \begin{cases} |g^*(\mathbf{x}_i)|, & \text{if} \quad g^*(\mathbf{x}_i) < 0 \\ 0, & \text{otherwise} \end{cases}$$

All constraint violations are then added together:

$$\varOmega(\mathbf{x}_i) = \sum_{j=1}^{n_g} \omega_m(\mathbf{x}_i)$$

The combined constrained violation value ($\varOmega$) is multiplied with a penalty parameter (P_k) and this product is then added to each of the objective function values as follows:

$$f^*_k(\mathbf{x}_i) = f_k(\mathbf{x}_i) + P_k \varOmega(\mathbf{x}_i) \tag{2}$$

For a feasible solution the $\varOmega$ term in Eq. (2) will be zero and $f^*_k = f_k$.

The second approach is to keep the objective function (f_k) unchanged, but to modify the Pareto-dominance comparison of vectors. When solving a MOP a single fitness value does not exit. Therefore, a new approach is required to compare two solutions' quality. Typically, Pareto-dominance is used:

Definition 1. *Assuming minimization, a solution $\mathbf{x}_i$ is better in quality than another solution $\mathbf{x}_j$ if:*

$$f(x^a_i) < f(x^a_j) \ \ \forall a = 1, \ldots, A \tag{3}$$

$$\exists \ x^b_i : f(x^b_i) \leq f(x^b_j) \ \ \textit{with} \ \ b \in [1, n_x] \tag{4}$$

If $\mathbf{x}_i$ is better than $\mathbf{x}_j$, $\mathbf{x}_i$ dominates $\mathbf{x}_j$, written as $\mathbf{x}_i \prec \mathbf{x}_j$. If not one of the solutions are better in quality than the other, the solutions are non-dominated. The set of non-dominated solutions in the objective space is referred to as the Pareto-optimal front (POF).

Definition 1 does not take constraint violations into consideration. Therefore, when comparing solutions of a CMOP Definition 1 has to be adapted as follows:

Definition 2. *Assuming minimization, a solution $\mathbf{x}_i$ constrain-dominates another solution $\mathbf{x}_j$ if:*

1. $\mathbf{x}_i$ is feasible and $\mathbf{x}_j$ is infeasible
2. $\mathbf{x}_i$ and $\mathbf{x}_j$ are both infeasible, but $\mathbf{x}_i$ has a smaller constraint violation
3. $\mathbf{x}_i$ and $\mathbf{x}_j$ are both feasible and $\mathbf{x}_i \prec \mathbf{x}_j$ according to Definition 1.

2.2 Algorithms

The non-dominated sorting genetic algorithm II (NSGA-II) is a multi-objective genetic algorithm (MOGA) modelled on genotypic behaviour and is still the standard benchmark algorithm in multi-objective optimization (MOO) studies [7,14]. The algorithm improves on the $O(MN^3)$ complexity of previous non-dominated sorting genetic algorithm (GA) [21], as well as adds elitism whilst removing the need for a sharing parameter [7].

PAES is a MOEA that uses an (1+1)-ES that uses only mutation on a single parent to create a single offspring. Therefore, PAES is a local search strategy [13]. It uses an archive to store the non-dominated solutions found so far and it maintains diversity through a crowding procedure that recursively divides the objective space into a grid.

MOEA/D is an EA based on the concept of decomposition [22]. It decomposes a MOP into a number of sub-problems and then optimizes the sub-problems simultaneously. Each sub-problem is optimized using only information from its neighbouring sub-problems. Each population contains the best solution for each of the sub-problems.

EAs mainly focus on genetic concepts and natural selection. However, CAs model cultural evolution and are based on theories in sociology and archeology [19]. Individuals are described through behavioural traits and can generate generalized descriptions of their experience. CA was extended to solve MOPs by Coello and Becerra [4].

3 Experimental Setup

This section discusses the experimental setup used for this study. The algorithms are discussed in Sect. 3.1. Sections 3.2 and 3.3 present the benchmarks and performance measures that were used to evaluate the performance of the algorithms on CMOPs. The statistical analyses conducted on the obtained results are discussed in Sect. 3.4.

3.1 Algorithms

The following algorithms were used in this study: SNSGA-II [7], PAES [13], MOEA/D [22] and CA [4]. For each of these algorithms two configurations were used in the study, namely one configuration that implemented the penalty function approach and one configuration that implemented the modified Pareto-dominance approach (refer to Sect. 2.1).

Each algorithm was run 30 times for a maximum of 25 000 fitness evaluations. Each algorithm had a population size of 100 and PAES used 3 bi-sections. The crossover probability was set to 0.75 and the mutation probability to 0.1, since these values promote a balance between exploration and exploitation of the search space [9]. Simulated binary crossover (SBX) [6] and polynomial mutation [5] were used to generate the offspring, where applicable. For MOEA/D, the neighbour size was set to 20, the neigbour selection probability to 0.9 and the maximum number of replaced solutions to 2.

3.2 Benchmark Functions

The following benchmark functions were used to evaluate the performance of the various algorithms:

- Binh2 with 2 decision variables and 2 inequality constraints [1], referred to as f_1
- Osyczka2 that has 6 decision variables and 6 inequality constraints [15], referred to as f_2
- Srinivas with 2 decision variables and 2 inequality constraints [21], referred to as f_3
- Two Bar Truss with 3 decision variables, 4 inequality constraints and a boundary constraint [12], referred to as f_4
- Welded Beam that has 4 decision variables and 4 constraints [17], referred to as f_5

3.3 Performance Measures

This section discusses the performance measures that were used to evaluate the performance of the MOEAs.

Inverted Generational Distance (IGD). Inverted generational distance (IGD) measures the distance between each solution in the optimal POF and the closest solution in the approximated POF [20]. A small value indicates that an algorithm performed well with regards to both convergence and diversity.

Hypervolume (HV). The hypervolume (HV) or S-metric measures how much of the objective space is dominated by a non-dominated set [23,24]. The reference vector or reference point that was used in the HV calculation was the vector that consisted of the worst value for each objective of the union of all non-dominated solutions of all approximated POFs that were compared against each other. A high HV value indicates good performance.

ϵ Metric (ϵ_m). Zitzler *et al.* presented the ϵ-metric (ϵ_m) to compare approximated sets [25]. It measures the factor by which an approximation set is worse than another approximation set with respect to all objectives, i.e. it provides the factor ϵ where for any solution in set B there is at least one solution in set A that is not worse by a factor of ϵ in all objectives. A small EP value indicates good performance.

3.4 Statistical Analysis

For each performance measure a Friedmann Test was performed to determine whether there was a statistical significant differences between the entire set of results. If the Friedmann Test indicated a statistical significant difference, pairwise Mann-Whitney U tests were conducted between the various algorithm configurations. If the Mann-Whitney U test indicated a statistical significant difference, the algorithm with the best average for the specific measure was awarded a win and the other algorithm a loss. All statistical tests were conducted with a confidence level of 95%.

4 Results

This section discusses the results that were obtained from the experiments.

4.1 Inverted Generational Distance

This section discusses the IGD values that were obtained by the various algorithms. The IGD mean and standard deviation values are presented in Table 1. In Table 1 D refers to the modified Pareto-dominance approach and P refers to the penalty function approach.

Table 1. IGD mean and standard deviation values

	SSNSGA-II(D)	SSNSGA-II(P)	PAES(D)	PAES(P)
f_1	1.17E−04 ± 2.17E−06	1.18E−04 ± 1.68E−06	1.19E−02 ± 1.86E−03	1.16E−02 ± 1.93E−03
f_2	4.09E−03 ± 1.19E−04	4.06E−03 ± 1.53E−04	2.63E−02 ± 1.91E−02	3.08E−02 ± 1.67E−02
f_3	8.14E−04 ± 3.55E−06	7.02E−04 ± 3.74E−06	1.31E−02 ± 1.01E−02	1.77E−02 ± 8.08E−03
f_4	7.05E−04 ± 6.20E−06	5.98E−04 ± 1.28E−05	**1.48E−02 ± 7.02E−03**	1.65E−02 ± 7.02E−03
f_5	2.05E−03 ± 1.06E−06	2.03E−03 ± 3.47E−05	2.23E−02 ± 9.88E−03	1.85E−02 ± 1.06E−02
	MOEA/D(D)	MOAE/D(P)	CA(D)	CA(P)
f_1	5.08E−04 ± 3.72E−06	5.18E−04 ± 7.91E−06	**1.07E−02 ± 1.37E−03**	1.19E−02 ± 1.79E−03
f_2	4.84E−03 ± 3.41E−04	**4.48E−03 ± 2.55E−04**	5.40E−02 ± 2.79E−02	4.95E−02 ± 3.14E−02
f_3	8.19E−04 ± 8.87E−07	7.20E−04 ± 1.85E−06	1.48E−02 ± 8.48E−03	2.00E−02 ± 7.86E−03
f_4	1.02E−02 ± 1.72E−04	**1.01E−02 ± 1.62E−04**	1.28E−02 ± 3.56E−03	1.32E−02 ± 2.60E−03
f_5	1.27E−02 ± 4.07E−04	**1.10E−02 ± 2.93E−04**	**7.66E−02 ± 8.49E−02**	2.65E−01 ± 4.79E−01

From Table 1 it can be seen that for SNSGA-II both constraint-dealing approaches performed equally well. Not one of the approaches obtained a better average and standard deviation value for any of the functions. For MOEA/D the penalty function approach obtained the best IGD average and standard deviation values for 3 of the functions (indicated in bold) and the modified Pareto-dominance approach for none of the functions. In contrast, for the CA and PAES, the modified Pareto-dominance approach obtained the best IGD average and standard deviation values for 2 and 1 of the 5 functions respectively, and the penalty function approach for none of the functions.

Applying a Friedman Test on the means of the IGD values revealed that the null hypothesis had to be rejected, and hence, there was a statistical significant difference in the performance of the algorithms. The results of the Mann-Whitney U tests are presented in Table 2. In Table 2, < indicates that the left hand algorithm performed statistical significantly better than the right hand algorithm, > indicates that the left hand algorithm performed statistical significantly worse than the right hand algorithm, and = indicates equivalent performance. The

Table 2. Comparison of algorithms' performance using Mann-Whitney U tests

	SSNSGA-II(D)	SSNSGA-II(P)	PAES(D)	PAES(P)	MOEA/D(D)	MOAE/D(P)	CA(D)	CA(P)
SSNSGA-II(D)	N/A	=	<	<	=	=	<	<
SSNSGA-II(P)	=	N/A	<	<	=	=	<	<
PAES(D)	>	>	N/A	=	>	>	=	=
PAES(P)	>	>	=	N/A	=	>	=	=
MOEA/D(D)	=	=	<	=	N/A	=	<	<
MOAE/D(P)	=	=	<	<	=	N/A	<	<
CA(D)	>	>	=	=	>	>	N/A	=
CA(P)	>	>	=	=	>	>	=	N/A

results indicate that for each of the MOEAs there was not a statistical significant difference in performance between the configurations that incorporated the two constraint-dealing approaches. However, there was a statistical significant difference in performance when the various MOEAs were compared against one another. Both SNSGA-II configurations outperformed both configurations of PAES and CA. MOEA/D using the penalty function also outperformed both configurations of PAES and CA. The modified Pareto-dominance MOEA/D outperformed both configurations of CA and only the modified Pareto-dominance approach of PAES. The modified Pareto-dominance MOEA/D experienced no statistical significant difference in performance with the penalty function configuration of PAES. There was also no statistical significant difference in performance between the SNSGA-II and MOEA/D configurations.

The percentage of wins obtained by each of these two constraint-dealing approaches for IGD is presented in Table 3. The results indicate that the penalty function approaches marginally outperformed the modified Pareto-dominance approaches.

Table 3. Wins obtained by the two constraint-dealing approaches

	D	P
Win percentage	12.5%	19.64%

4.2 Hypervolume and ϵ-metric

Friedman Tests on the mean HV values and the mean ϵ_m values indicated that there was no statistical significant difference in performance of the various algorithms. The HV and ϵ_m values are presented in Tables 4 and 5.

The penalty-based SNSGA-II outperformed the modified Pareto-dominace SNSGA-II by obtaining the better HV mean and deviation for 2 functions.

Table 4. HV Mean and standard deviation values

	SSNSGA-II(D)	SSNSGA-II(P)	PAES(D)	PAES(P)
f_1	**8.12E−01 ± 2.76E−05**	8.12E−01 ± 4.38E−05	4.52E−01 ± 7.61E−02	4.64E−01 ± 9.86E−02
f_2	1.19E−01 ± 1.20E−03	1.20E−01 ± 1.75E−03	1.20E−01 ± 8.57E−02	**1.63E−01 ± 7.88E−02**
f_3	3.92E−01 ± 1.93E−04	**4.90E−01 ± 1.01E−04**	2.10E−01 ± 1.13E−01	1.46E−01 ± 1.13E−01
f_4	9.19E−01 ± 4.32E−04	9.09E−01 ± 2.42E−04	4.49E−01 ± 2.12E−01	4.98E−01 ± 2.41E−01
f_5	9.46E−01 ± 1.74E−03	**9.56E−01 ± 1.50E−03**	**7.90E−01 ± 2.84E−01**	6.57E−01 ± 2.92E−01
	MOEA/D(D)	MOAE/D(P)	CA(D)	CA(P)
f_1	**8.08E−01 ± 2.70E−05**	8.08E−01 ± 6.32E−05	**4.53E−01 ± 7.85E−02**	4.09E−01 ± 8.72E−02
f_2	1.00E−01 ± 6.96E−03	9.76E−02 ± 6.60E−03	1.49E−02 ± 3.39E−02	**1.87E−02 ± 3.22E−02**
f_3	3.91E−01 ± 1.09E−04	4.87E−01 ± 1.82E−04	**1.28E−01 ± 8.13E−02**	9.84E−02 ± 1.08E−01
f_4	1.00E−01 ± 1.12E−02	**1.08E−01 ± 1.10E−02**	2.35E−01 ± 1.90E−01	4.41E−01 ± 1.99E−01
f_5	9.48E−01 ± 7.17E−06	**9.50E−01 ± 6.61E−04**	**2.49E−01 ± 2.86E−01**	1.63E−01 ± 1.85E−01

Table 5. ϵ_m mean and standard deviation values

	SSNSGA-II(D)	SSNSGA-II(P)	PAES(D)	PAES(P)
f_1	**5.67E−03 ± 4.22E−04**	5.90E−03 ± 9.40E−04	4.68E−01 ± 1.23E−01	4.43E−01 ± 1.38E−01
f_2	**8.62E−01 ± 2.99E−03**	8.62E−01 ± 3.99E−03	8.08E−01 ± 1.59E−01	**7.60E−01 ± 1.19E−01**
f_3	5.36E−01 ± 2.94E−04	**3.69E−01 ± 1.96E−04**	6.94E−01 ± 1.99E−01	8.08E−01 ± 1.81E−01
f_4	7.20E−02 ± 4.66E−04	**6.18E−02 ± 3.22E−04**	5.47E−01 ± 2.15E−01	4.73E−01 ± 2.66E−01
f_5	5.43E−02 ± 1.74E−03	**4.16E−02 ± 1.55E−03**	**2.10E−01 ± 2.84E−01**	3.39E−01 ± 2.95E−01
	MOEA/D(D)	MOAE/D(P)	CA(D)	CA(P)
f_1	2.12E−02 ± 1.27E−04	**2.12E−02 ± 1.15E−04**	**4.14E−01 ± 9.56E−02**	5.07E−01 ± 1.25E−01
f_2	8.53E−01 ± 9.22E−12	8.54E−01 ± 3.74E−12	**1.20E+00 ± 3.35E−01**	1.21E+00 ± 5.17E−01
f_3	5.35E−01 ± 1.80E−14	3.69E−01 ± 7.32E−12	**8.10E−01 ± 1.57E−01**	9.91E−01 ± 2.84E−01
f_4	9.00E−01 ± 1.12E−02	**8.92E−01 ± 1.10E−02**	8.18E−01 ± 3.09E−01	**5.39E−01 ± 2.60E−01**
f_5	5.16E−02 ± 7.39E−06	4.07E−02 ± 1.27E−03	**1.97E+00 ± 1.98E+00**	6.01E+00 ± 1.04E+01

A similar trend was observed for MOEA/D. For the CA, the modified Pareto-dominance approach outperformed the penalty-based approach on 3 of the 5 functions. However, both approaches performed equally well for PAES. A similar trend was observed for ϵ_m.

5 Conclusion

This study investigated the effect of two constraint-dealing approaches on the performance of the steady-state non-dominated sorting genetic algorithm II (SNSGA-II), the Pareto-archived evolution strategy (PAES), the multi-objective evolutionary algorithm based on decomposition (MOEA/D) and a cultural algorithm (CA). The two constraint-dealing approaches are: adding a penalty term to each objective function; and using a modified Pareto-dominance approach that incorporates constraint violations. Each multi-objective evolutionary algorithm (MOEA) had two configurations, where each configuration incorporated one of these approaches to deal with constraints.

The results indicated that there was no statistical significant difference between the two constraint-dealing approaches. However, the penalty function approach did slightly outperform the modified Pareto-dominance approach based on the wins and losses. Furthermore, for the CA the modified Pareto-dominance approach performed better on all performance measures. In contrast, the penalty-based MOEA/D outperformed the modified Pareto-dominance version on all performance measures. For PAES, both approaches performed equally well, except on the inverted generational distance (IGD) measure where the modified Pareto-dominance approach slightly outperformed the penalty function approach. The penalty function SNSGA-II slightly outperformed the modified Pareto-dominance SNSGA-II on the hypervolume (HV) and ϵ_m. However, both approaches performed equally well on IGD.

A penalty function modifies the objective space and can lead to changes in the optima. The modified Pareto-dominance approach does not change the objective functions. Since there was not a huge statistical significant difference in performance between these two approaches, it is simpler to rather use the modified Pareto-dominance approach than to add a penalty function where the ideal penalty parameter values, that are problem dependent, should be found.

Future work will include extending the study to incorporate additional multi-objective optimization constrained problems [2], as well as adaptive penalty approaches [10].

References

1. Binh, K.: MOBES: A multi-objective evolution strategy for constrained optimization problems. In: Proceedings of the International Conference on Genetic Algorithm, Brno, Czech Republic, pp. 176–182 (1997)
2. Cheng, P.: A tunable constrained test problems generator for multi-objective optimization. In: Proceedings of the International Conference on Genetic and Evolutionary Computing, Washington, DC, USA, pp. 96–100, September 2008
3. Coello Coello, C.A.: Evolutionary multi-objective optimization: a historical view of the field. IEEE Comput. Intell. Mag. **1**(1), 28–36 (2006)
4. Coello Coello, C.A., Becerra, R.L.: Evolutionary multiobjective optimization using a cultural algorithm. In: Proceedings of the IEEE Swarm Intelligence Symposium, pp. 6–13 (2003)
5. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms, vol. 16. Wiley, New York (2001)
6. Deb, K., Beyer, H.G.: Self-adaptive genetic algorithms with simulated binary crossover. Technical report CI-61/99, University of Dortmund (1999)
7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
8. Edgeworth, F.: Mathematical Physics. P Keagan, London (1881)
9. Eiben, A.E., Hinterding, R., Michalewicz, Z.: Parameter control in evolutionary algorithms. IEEE Trans. Evol. Comput. **3**(2), 124–141 (1999)
10. Harada, K., Sakuma, J., Ono, I., Kobayashi, S.: Constraint-handling method for multi-objective function optimization: pareto descent repair operator. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 156–170. Springer, Heidelberg (2007). doi:10.1007/978-3-540-70928-2_15

11. Jiménez, F., JM Cadenas, J.: Evolutionary techniques for constrained multiobjective optimization problems. In: Proceedings of the Workshop on Multi-Criterion Optimization Using Evolutionary Methods: Genetic and Evolutionary Computation Conference, pp. 115–116 (1999)
12. Kirsch, U.: Optimal Structural Design. McGraw-Hill, New York (1981)
13. Knowles, J., Corne, D.: The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation. In: Proceedings of the IEEE Congress on Evolutionary Computation, vol. 1 (1999)
14. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. IEEE Trans. Evol. Comput. **13**(2), 284–302 (2009)
15. Osyczka, A., Kundu, S.: A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. Struct. Optim. **10**, 94–99 (1995)
16. Pareto, V.: Cours D'Economie Politique. F. Rouge, Lausanne (1896)
17. Ragsdell, K., Phullips, D.: Optimal design of a class of welded structures using geometric programming. J. Eng. Ind. Ser. B **98**, 1021–1025 (1975)
18. Ray, T.: An evolutionary algorithm for multiobjective optimization. Eng. Optim. **33**(3), 399–424 (2001)
19. Reynolds, R.: An introduction to cultural algorithms. In: Proceedings of the Annual Conference on Evolutionary Programming, pp. 131–139 (1994)
20. Sierra, M., Coello Coello, C.: Improving PSO-Based multi-objective optimization using crowding, mutation and ϵ-dominance. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 505–519. Springer, Heidelberg (2005)
21. Srinivas, N., Deb, K.: Multi-objective function optimization using non-dominated sorting genetic algorithms. Evol. Comput. J. **2**(3), 221–248 (1994)
22. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007)
23. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans. Evol. Comput. **3**(4), 257–271 (1999)
24. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms — a comparative case study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 292–301. Springer, Heidelberg (1998). doi:10.1007/BFb0056872
25. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans. Evol. Comput. **7**(2), 117–132 (2003)