CYLINDRICAL CONSTRAINT EVOLUTIONARY ALGORITHM FOR MULTIOBJECTIVE OPTIMIZATION

Tohid Erfani and Sergei V. Utyuzhnikov
The University of Manchester, School of Mechanical, Aerospace and Civil Engineering
George Begg Building, M13 9PL, Manchester, U.K.

Keywords: Cylindrical constraint method, Multiobjective optimization, Evolutionary optimization, Evolutionary algorithms, Genetic algorithm.

Abstract: This paper introduces a new iterative evolutionary algorithm, which is able to provide an evenly distributed set of solutions in multiobjective context. The method is different from the other evolutionary algorithms in two perspectives. First, instead of density information incorporated to find a diverse set of solutions, a hypercylinder is introduced as a new constraint to the problem. Searching for the solution within this hypercylinder guarantees the evenly generated solutions at the end of the optimization process. Second, a fitness function is constructed to handle the problem constraints and meanwhile minimize the distance of the solution to the true optimum frontier. In addition, the method is developed in such a way that it can be easily implemented in searching the preferable region of the search space. The algorithm behaviour is tested on different test cases and the results are compared in both convergence and diversity to those of other well known approaches to demonstrate the efficacy of the proposed method.

1 INTRODUCTION

Most of the optimization problems in the real world are inherently multiobjective. That is, for a single problem, there exists possibly conflicting objectives which need to be optimized simultaneously. This leads to generating a so called *trade-off* surface. The solution on this frontier is called a *Pareto* optimal solution in a sense that no improvement can be made with respect to one objective without deterioration of at least one of the other. In the literature, there are two principle approaches to solving such problems; *classical directional* (gradient) based methods and *evolutionary* algorithms. In the first category, all the objective functions are aggregated to form a single objective optimization sub-problem. Different sub-problems are then solved for a different position in the search space to obtain the Pareto frontier. Normalized Normal Constraint method (NNC) (Messac et al., 2003) and Directed Search Domain method (DSD) (Erfani and Utyuzhnikov, 2010) are amongst the directional methods capable in generating the whole Pareto surface. In the latter category, the focus is to get the whole trade-off surface in a single optimization run using evolutionary techniques such as NSGA-II (Deb et al., 2002) and SPEA2 (Zitzler et al., 2001) as two well-studied algorithms. Generally, there are two issues which need to be accomplished in both categories. Firstly, the generated solutions should have minimum distance to the true trade-off surface and secondly, a diverse set of solutions should be obtained, which guarantees a good approximation of the whole Pareto frontier. The latter issue is specially important in engineering design, where the designer is only capable in analysing a very limited set of solutions.

With respect to the first task, the directional methods utilize the gradient based search for any single objective optimization sub-problem to move towards the optimal solution quickly and efficiently (Jahn, 2004). However, since these methods need the starting point to initialize the sub-problem search, different starting values may result in different solutions (Boyd and Vandenberghe, 2004). That is, the possibility of getting stuck to a local optimum brings an added difficulty to the methods. On the other hand, in the evolutionary techniques, with different detailed procedure for different algorithms, a mating selection is used to mainly rank the individuals based on the Pareto definition (Zitzler et al., 2001). For example, in SPEA2 the the strength of a solution is defined by the number of solutions it dominates (Zitzler et al., 2001). In

Erfani T. and V. Utyuzhnikov S..
CYLINDRICAL CONSTRAINT EVOLUTIONARY ALGORITHM FOR MULTIOBJECTIVE OPTIMIZATION.
DOI: 10.5220/0003670101840189
In *Proceedings of the International Conference on Evolutionary Computation Theory and Applications* (ECTA-2011), pages 184-189
ISBN: 978-989-8425-83-6

NSGA-II, the solutions are ranked in different frontiers based on the Pareto definitions (Deb et al., 2002). In both cases, optimizing the rank and strength of the solutions, so far found by the algorithms, assures the proximity of the current solution to the Pareto frontier.

The second task is also tackled differently by these two classes of methods. In the directional methods, a so called utopia plane is formed by an evenly distributed set of points (Das and Dennis, 1998). The aim is to achieve an evenly distributed set of Pareto solutions by solving numerous sub-problems corresponding to each point. However, as shown by (Das and Dennis, 1998; Messac et al., 2003; Erfani and Utyuzhnikov, 2010), the points on utopia plane may not cover the whole Pareto surface and hence loosing some portion of solutions. To alleviate the inadequate extent of the utopia plane, specifically NNC and DSD methods utilize the *extended* utopia plane and *rotation* technique, respectively. However, in the case of degeneration of the utopia plane to lower dimension, NNC technique may not sound promising. In the evolutionary algorithms, on the other hand, the *density* information is exploited to define an extra order between the individuals (Zitzler and Thiele, 2002). In SPEA2, the *k*-th nearest neighbour method is used to consider the density information of the individuals to finally discard the solutions which are so close to each other. In NSGA-II, a *crowding* distance is defined, which is the average distance of the two points in either side of an individual for each objective function. This measure is then used to avoid a dense of solutions in one part of the frontier in the case that some other parts are isolated (Deb et al., 2002).

In this study we introduce a method which incorporates the desired strength of the evolutionary and directional methodology in order to obtain a new algorithm. In particular the main characteristics of the method are:

First. To generate an evenly distributed set of solution, a hyperplane is constructed with evenly distributed points on it. Afterwards, a *hypercylinder* constraint is defined for each point which its feasibility in the optimization process guarantees the evenly generated solution. This removes the burden of sorting and clustering techniques for ranking the solutions based on their density information.

Second. With the mutation and recombination remain identical to those introduced in literature (Haupt et al., 1998), a fitness function and simple sampling strategy is introduced to fill up the mating pool for selection.

2 CYLINDRICAL CONSTRAINT EVOLUTIONARY ALGORITHM

Our approach is based on generating an evenly distributed set of points m on a plane P defined by the vectors of the Cartesian coordinate system $\hat{e}_i$ (e.g. $\hat{e}_1 = (1,0)$ and $\hat{e}_2 = (0,1)$ for 2-objective case):

$$\begin{aligned} P &= \sum_{i=1}^{n} \alpha_i \hat{e}_i, \\ \sum_{j=1}^{n} \alpha_j &= 1, \\ 0 \leq \alpha_j &\leq 1. \end{aligned} \tag{1}$$

In contrast to the other directional techniques introduced in literature (Messac et al., 2003; Erfani and Utyuzhnikov, 2010), we do not use the utopia hyperplane as it can not cover the whole Pareto surface. Instead, we define a direction from the utopia point U^* (the point which its components are the minimum of each objective in each generation) to each m by

$$\nu = m - U^*. \tag{2}$$

This represents a set of rays emitting from U^* towards the m on polygon P. Associated with each m, a scalar optimization problem is solved. The point m for each scalar problem serves as the problem constraint defined by a cylinder. The cylinder is constructed by a circle on a plane extended in direction

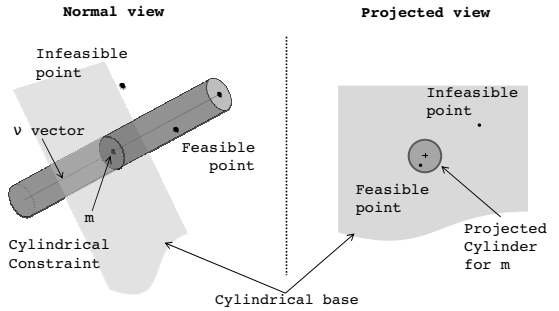


Figure 1: Explaining the CCEA method for a given point m

of ν (Look at Figure 1). Following the Figure 1, one possible way to check whether the cylinder constraint is satisfied is to project the current searching point on the cylinder base and investigate whether or not the projected point is inside the current circle centred on m with predefined radius r. The cylinder base is a plane with ν as its normal vector. Since the $m \in P$ are evenly generated, also the cylinders are and hence (quasi) evenly distributed set of solutions are guaranteed by solving the optimization in each cylinder.

The Cylindrical Constraint Evolutionary Algorithm (CCEA) is given below. In the first steps, the

problem is initialized and individuals from *pop* is assigned to each m on P. This makes the size of the *pop* to be the same as the number of the points m on P. Therefore, each individual with dedicated point m_i is updated during the optimization of sub-problem i. In addition, the utopia point U^* is computed with respect to the current population in objective space.

During the course of optimization of sub-problem i corresponds to point m_i, two points from *pop* are randomly selected and *binary tournament* selection is applied. The resultant p_b from tournament is then reproduced using genetic operators on p_b and p_m. The offspring p_{re} is checked for its fitness value. If it performs better, it replaces the p_m. In addition to this, we have introduced one further checking. If the offspring p_{re} does not perform better with respect to sub-problem i, it may be a better solution for the other sub-problems. That is, the genetic operators may produce a solution near another cylinder which may perform better with respect to that cylinder. Therefore, we check the fitness of p_{re} for the possible better performance with respect to the other sub-problems. Since the utopia point is changing from generation to generation, its value is updated using the new objective values of the updated individuals.

To terminate the optimization procedure, it is recommended to follow two rules. If for point m_i, the average value of the best fitness function for a given number of generation is unchanged with a prescribed accuracy, the search in the cylinder i will be stopped while its solution will still be used for the other sub-problems' reproduction and tournament selection purposes. However, if the maximum generation is reached, the optimization procedure will be terminated. Once the termination criteria is satisfied, a Pareto filtering is implemented. The filtering aims to eliminate the local Pareto solutions based on the non-domination definition. One may find such a filtering in (Erfani and Utyuzhnikov, 2010).

The tournament selection and fitness checking steps described above are based on constructing the following fitness function. The fitness function for sub-problem i is defined with two separate terms in order of their preference and importance:

$$Fitness_i = \alpha f_1 + \beta f_2. \quad (3)$$

To define f_1, we first describe the cylindrical constraint. Let $F_c = F(x_c)$ be the current searching point in objective space and F_c^p be its orthogonal projection on i-th cylinder base. Then,

$$d(F_c^p, \nu_i) \leq r \quad (4)$$

is the cylindrical constraint for sub-problem i. $d(,)$ is the Euclidean distance between two points and r is the predefined radius of the cylinder. To guarantee one distinct solution in each cylinder, r is suggested to be less than a half of the distance between any two neighbour points m generated on P. To incorporate this constraint into the fitness function, f_1 is defined as

$$f_1 = max(0, d(F_c^p, \nu_i) - r). \quad (5)$$

f_1, is the penalty function measuring the amount of violation for all the constraints (if there is any) including the cylinder one. One may calculate the penalty to be larger outside the feasible space by multiplying the f_1 by itself. Although some other constraint handling techniques may be useful, the proposed penalty function seems to work reasonably for our purpose.

The second term in Equation 3, f_2, is introduced by

$$f_2 = d(F_c, U^*), \quad (6)$$

which is the distance of the individuals from the utopia point U^*. Since U^* is situated outside the feasible space, minimizing the f_2 guarantees the minimum distance to the true Pareto frontier for minimization problems. The order of the preference in Equation 3 can be incorporated by introducing a weight for each term. We propose the higher weight for the first term, α, to assure that the solution is obtained as closely as possible inside the cylinder to guarantee the final evenly generated set of Pareto solutions.

3 EXPERIMENTAL TEST SETUPS

In this section we study the behaviour of the CCEA method on the ZDT and DTLZ selected problems introduced in (Deb et al., 2005). The individuals are coded as real value vectors. The population size is set to n =52 for both two- and three-dimensions problems with *MaxGen* =50 to deal with lower computational cost. This leads to 2600 fitness evaluations for each test problem.

For the matter of comparison with CCEA, we have adopted NSGA-II and SPEA2 algorithms. In these three algorithms with the same genetic operators, we set crossover rate as 1 and mutation rate as $1/k$ where k is the number of decision variables.

3.1 Performance Assessment

To evaluate the performance of the CCEA algorithm on the test cases, 10 independent algorithm runs are carried out. To account for both diversity and quality (divergence) of the solutions, the hypervolume indicator(Deb, 2001) is computed. Hypervolume metric estimates the volume (in the objective space) covered by the individuals in a given solution set. For

our case, we have used hypervolume metric to calculate the ratio between the volume of obtain Pareto solutions (PA_o) and the volume of true Pareto frontier (PA_f) bounded by a reference point as

$$\Delta = \frac{H(PA_o, ref)}{H(PA_f, ref)}. \quad (7)$$

The reference points are computed based on the worst value of each objective function for a given test case using all three algorithms. Given the PA_f as the true set of Pareto solutions, Δ for a perfect situation equals 1. Therefore the closer the Δ to 1, the better the spread and divergence of solutions. For our purpose $|PA_f|$=300 and 500 evenly distributed points are sampled from the Pareto frontier for two- and three objective problems, respectively.

In the Figures, the best value recorded in 10 runs are shown as the result. Table 1 records the Δ metric with the variance of the 10 algorithm runs for different test cases.

3.2 Test Problems and Discussion of Results

For the formulation of the test cases, the reader may refer to (Deb et al., 2005). All the test cases have k =10 decision variables. In all of them, the true Pareto frontier is visualized in order to make the assessment easier.

Table 1: Comparison between CCEA, NSGA-II and SPEA2 on ZDT test cases using hypervolume (Δ) metric. (**bold** shows better performance on average) The figures in parentheses are the variance.

	Δ metric on average of 10 independent run for ZDTs		
	ZDT1	ZDT3	ZDT6
NSGAII	0.962 (*0.013*)	0.971 (*0.011*)	0.817 (*0.092*)
SPEA2	0.933 (*0.033*)	0.763 (*0.044*)	0.751 (*0.052*)
CCEA	**0.996** (*0.003*)	**0.991** (*0.008*)	**0.995** (*0.004*)
Ref.	[2,2]	[2,4]	[2,10]

ZDT1. Figure 2 demonstrates the generated points using CCEA algorithm for this convex test case. In addition, in Table 1 , the Δ metric shows that estimated solutions using the CCEA has better approximation than those produced by NSGA-II and SPEA2. As demonstrated visually in Figure 2, it is evident that the CCEA outperforms the other two methods with small number of generation and population.

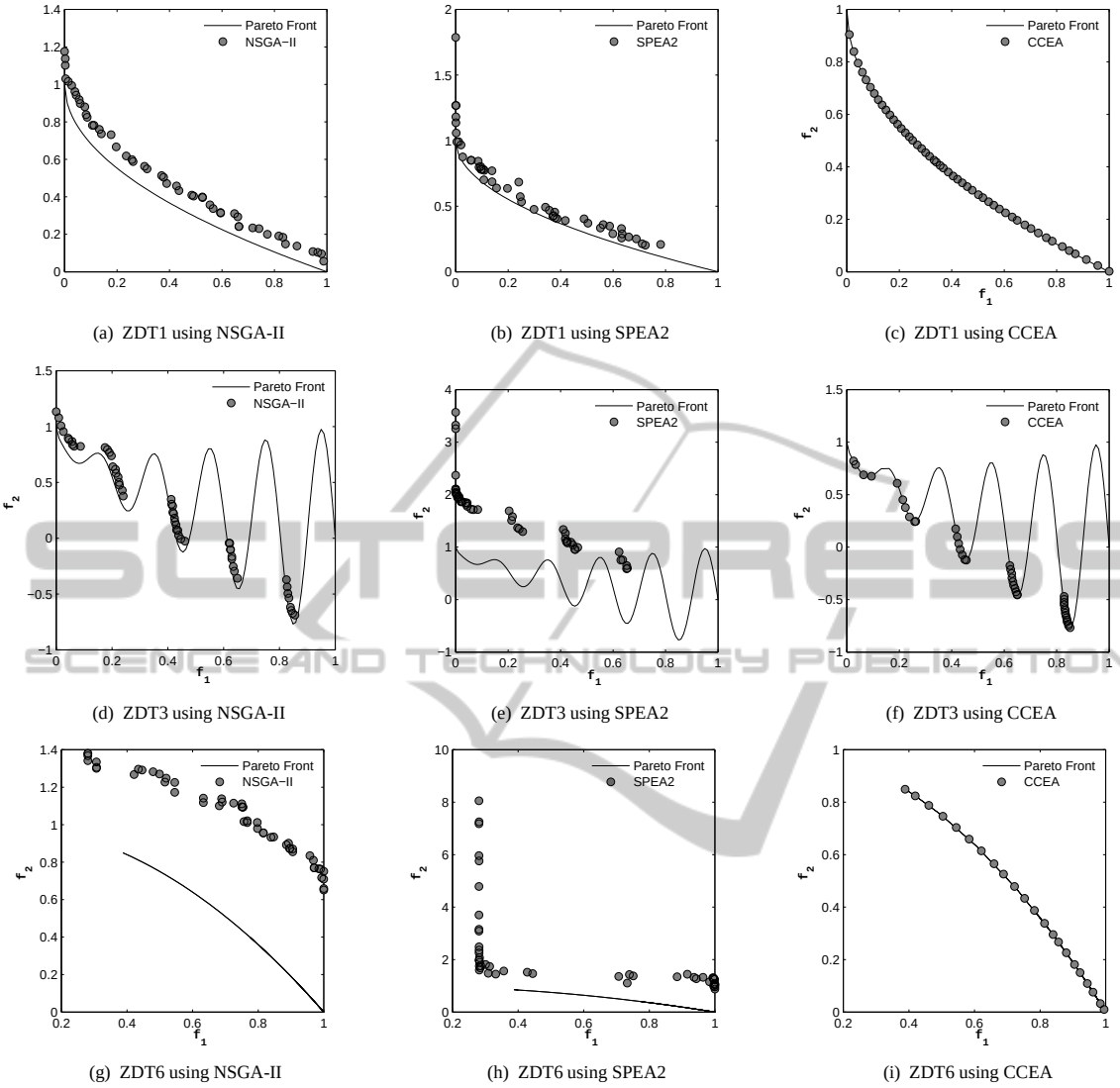
ZDT3 This test case is highly oscillated and has a disconnected Pareto frontier. Table 1 shows that the SPEA2 performs worst while NSGA-II obtained a reasonably better approximation. However, as can be seen in Figure 2, with the given number of generation, the algorithm fails to diverge perfectly to the Pareto frontier. Meanwhile, looking at Table 1, the results of CCEA indicate a diverse approximation of Pareto frontier as well as good quality of solutions with respect to divergence.

ZDT6. To account for the deceptive Pareto frontier, this test case has been adopted. Such a deceptive problems may encounter in real world optimization problems such as aerodynamic design task, where the fitness evaluation is also a costly job. Therefore, an efficient algorithm should be able to generate the whole Pareto surface with less computation cost. Having a non-convex Pareto set, the solutions across the Pareto boundary of this test case are non-uniformly distributed. As can be seen in Figure 2, NSGAII and SPEA2 can not maintain a good diverse set of solutions as well as reasonable divergence in 50 problem generations. CCEA, however, generates evenly distributed set of solution on the Pareto frontier demonstrated in Figure 2. Table 1 shows that the larger volume is captured using the approximation by CCEA compared to the other two methods.

DTLZ2. The Pareto surface in this test case is part of the unit sphere corresponding to non negative coordinates and is non-convex. By 52 different rays from utopia point (updated in each generation), we were able to cover the whole Pareto frontier without missing the peripheral areas; the difficulty faced in using directional classical methods. In addition, Table 2 and Figure 3 reveal that the results of CCEA is better in terms of quality and spread. Although NSGA-II were

Table 2: Comparison between CCEA, NSGA-II and SPEA2 on DTLZ test cases using hypervolume (Δ) metric. (**bold** shows better performance on average) The figures in parentheses are the variance.

	Δ metric on average of 10 independent run for DTLZs	
	DTLZ2	DTLZ5
NSGAII	0.923 (*0.032*)	**0.985** (*0.002*)
SPEA2	0.934 (*0.031*)	0.952 (*0.010*)
CCEA	**0.993** (*0.002*)	**0.984** (*0.005*)
Ref.	[2,2,2]	[2,2,2]

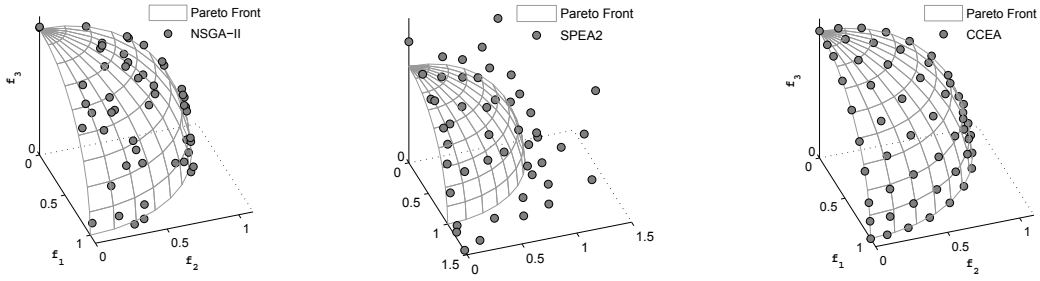


(a) ZDT1 using NSGA-II (b) ZDT1 using SPEA2 (c) ZDT1 using CCEA

(d) ZDT3 using NSGA-II (e) ZDT3 using SPEA2 (f) ZDT3 using CCEA

(g) ZDT6 using NSGA-II (h) ZDT6 using SPEA2 (i) ZDT6 using CCEA

Figure 2: Results of NSGA-II, SPEA2 and CCEA on 2-dimentional test problems .



(a) DTLZ2 using NSGA-II (b) DTLZ2 using SPEA2 (c) DTLZ2 using CCEA

Figure 3: Results of NSGA-II, SPEA2 and CCEA on DTLZ2 test problem .

able to approximate a reasonable diverse set of solutions, CCEA could manage to find a more evenly distributed set of solutions due to the cylindrical constraints application.

DTLZ5. The Pareto solutions for this problem lies on a curve as explained in (Deb et al., 2005). In addition, the anchor points which are used in classical algorithms to generate the utopia plane coincide each others and hence producing a problem for generating algorithms. As is evident in Table 2, The CCEA and NSGA-II perform pretty much the same in obtaining the Pareto curve.

4 CONCLUSIONS AND FUTURE WORK

In this paper, we introduced an evolutionary algorithm, CCEA, which is able to generate the whole Pareto frontier evenly distributed closely to the true Pareto surface. The method is based on introducing a uniformly distributed set of points on a plane and constructing a cylinder for each point. The axis of each cylinder is defined to be parallel to a vector connecting the utopia point to the corresponding point on the plane. The cylinder serves as the problem constraint, which assures the search to be biased towards its axis. By introducing a fitness function, the algorithm seems to reach near the Pareto optimal frontier while generating a diverse set of solutions. The experimental study shows that the solutions are reasonably distributed on the Pareto surface for a given test cases in literature. With regards to the computational cost while maintaining a good quality of diverse solution, the CCEA seems to outperform the NSGAII and SPEA2 on the given test cases. In addition and for further improvement, one may note that changing and tailoring the GA parameters may also speed up the performance of the CCEA algorithm. For future investigation, CCEA should be evaluated based on some other performance metrics and test cases. The performance of the method should also be investigated on higher dimension problems. In addition, a new development for computing the direction of the cylinders can increase the efficiency of the algorithm for some adversed scaled problems. Nevertheless, CCEA method can be used easily as a reference based technique to incorporate the preference into multiobjective optimization and design.

REFERENCES

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge Univ Pr.

Das, I. and Dennis, J. (1998). Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8:631.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Wiley.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2005). Scalable test problems for evolutionary multiobjective optimization. *Evolutionary Multiobjective Optimization*, pages 105–145.

Erfani, T. and Utyuzhnikov, S. (2010). Directed search domain: a method for even generation of the Pareto frontier in multiobjective optimization. *Engineering Optimization*, page DOI:10.1080/0305215X.2010.497185.

Haupt, R., Haupt, S., and Wiley, J. (1998). *Practical genetic algorithms*. Wiley Online Library.

Jahn, J. (2004). *Vector optimization: theory, applications, and extensions*. Springer Verlag.

Messac, A., Ismail-Yahaya, A., and Mattson, C. (2003). The normalized normal constraint method for generating the Pareto frontier. *Structural and Multidisciplinary Optimization*, 25(2):86–98.

Zitzler, E., Laumanns, M., Thiele, L., et al. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm. In *Eurogen*, volume 3242. Citeseer.

Zitzler, E. and Thiele, L. (2002). Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *evolutionary computation, IEEE transactions on*, 3(4):257–271.