A Survey of Decomposition Methods for Multi-objective Optimization

Alejandro Santiago, Héctor Joaquín Fraire Huacuja, Bernabé Dorronsoro, Johnatan E. Pecero, Claudia Gómez Santillan, Juan Javier González Barbosa and José Carlos Soto Monterrubio

Abstract The multi-objective optimization methods are traditionally based on Pareto dominance or relaxed forms of dominance in order to achieve a representation of the Pareto front. However, the performance of traditional optimization methods decreases for those problems with more than three objectives to optimize. The decomposition of a multi-objective problem is an approach that transforms a multi-objective problem into many single-objective optimization problems, avoiding the need of any dominance form. This chapter provides a short review of the general framework, current research trends and future research topics on decomposition methods.

A. Santiago · H. J. F. Huacuja (✉) · C. G. Santillan · J. J. González. Barbosa · J. C. S. Monterrubio
Instituto Tecnológico de Ciudad Madero, Ciudad Madero, Mexico
e-mail: automatas2002@yahoo.com.mx

A. Santiago
e-mail: alx.santiago@gmail.com

C. G. Santillan
e-mail: cggs71@hotmail.com

J. J. González. Barbosa
e-mail: jjgonzalezbarbosa@hotmail.com

J. C. S. Monterrubio
e-mail: soto190@gmail.com

B. Dorronsoro
University of Lille, Lille, France
e-mail: bernabe.dorronsoro_diaz@inria.fr

J. E. Pecero
University of Luxembourg, Luxembourg, Luxembourg
e-mail: johnatan.pecero@uni.lu

O. Castillo et al. (eds.), *Recent Advances on Hybrid Approaches for Designing Intelligent Systems*, Studies in Computational Intelligence 547,
DOI: 10.1007/978-3-319-05170-3_31,

1 Introduction

The decomposition of an optimization problem is an old idea that appears in optimization works [1–4]. This approach transforms the original problem into smaller ones and solves each problem separately. The decomposition could be at different levels: decision variables, functions and objectives. The decomposition in decision variables split in sub groups the original set of decision variables, and each group is optimized as a subproblem. Decomposition in decision variables is a novel technique in multi-objective optimization, works using this approach are found in Dorronsoro et al. [5] and Liu et al. [6]. The decomposition in functions can be done when the objective function can be decomposed into two or more objective functions. In (1) an example of a decomposable function is showed.

$$Minimize\, g(\vec{x}) = f_1(x_1, y) + f_2(x_2, y) \tag{1}$$

In this case, it is possible to optimize f_1 and f_2 in separately optimization problems [7–10]. One example in multi-objective optimization is the large-scale multi-objective non-linear programming problems [11]. It is important to notice that most of the multi-objective optimization problems cannot be optimized as individual optimization problems because their objective functions are not composed from subproblems. The decomposition in objectives is more natural than decomposition in functions for multi-objective optimization, given that every objective could represent a subproblem to optimize. This is the approach followed by the Nash genetic algorithms [12], however this approach finds solutions in equilibrium, not the optimal set. The coevolutionary algorithms can decompose in objectives, but it requires a different population for every objective [13]. The approach to decompose a multi-objective problem reviewed in this chapter is the one used in Zhang and Li [14]. The advantage of this approach is that it is not affected by the problem structure (decision variables, functions, number of objectives) and it is able to obtain a representation of the optimal set. The main contributions of our work can be summarized as follows:

- We identify the trends to decompose a multi-objective problem into single-objective problems and their weaknesses.
- We identify the general framework for multi-objective optimization by decomposition.
- We identify the current and future research trends for multi-objective decomposition.

The rest of the chapter is organized as follows. In Sect. 2, we present the basic concepts of multi-objective optimization and the concept of aggregate objective function. In Sect. 3, we present how the decomposition framework works and its advantages over based on dominance. We present the current research on multi-objective decomposition in Sect. 4. Section 5 provides our perspective on new research directions to be addressed in multi-objective decomposition. We conclude this chapter in Sect. 6.

2 Definitions and Background Concepts

This section presents the basic concepts in multi-objective optimization and the concepts used in multi-objective decomposition methods.

Definition 1 *Multi-objective optimization problem (MOP)*

Given a vector function $\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})]$ and its feasible solution space Ω, the MOP consists in find a vector $\vec{x} \in \Omega$ that optimizes the vector function $\vec{f}(\vec{x})$. Without loss of generality we will assume only minimization functions.

Definition 2 *Pareto dominance*

A vector $\vec{x}$ dominates $\vec{x}'$ (denoted by $\vec{x} \prec \vec{x}'$) if $f_i(\vec{x}) \leq f_i(\vec{x}')$ for all i functions in $\vec{f}$ and there is at least one i such that $f_i(\vec{x}) < f_i(\vec{x}')$.

Definition 3 *Pareto optimal*

A vector $\vec{x}^*$ is Pareto optimal if not exists a vector $\vec{x}' \in \Omega$ such that $\vec{x}' \prec \vec{x}^*$.

Definition 4 *Pareto optimal set*

Given a MOP, the Pareto optimal set is defined as $P^* = \{\vec{x}^* \in \Omega\}$.

Definition 5 *Pareto front*

Given a MOP and its Pareto optimal set P^*, the Pareto front is defined as $PF^* = \left\{\vec{f}(\vec{x}) | \vec{x} \in P^*\right\}$.

Aggregate functions use the concept of aggregate value to represent a set of individual values in a single one. Uses of the aggregate functions are found in: probability, statistics, computer science, economics, operations research, etc. [15]. In multi-objective optimization the aggregate objective function (AOF) transforms the functions in $\vec{f}$ into a single objective function. The simplest aggregate objective function used in multi-objective optimization is the following global objective sum.

$$Z = \sum\nolimits_{i=1}^{k} f_i(\vec{x}) \tag{2}$$

3 The Decomposition Framework

In this section we present the multi-objective decomposition approach, how to obtain Pareto optimal solutions, how to decompose a multi-objective optimization problem and the general decomposition framework.

Table 1 Common weighted aggregate objective functions

Name	Formulation	Designed for	Details in
Weighted sum	$\sum_{i=1}^{k} w_i f_i(\vec{x})$	Pareto optimality in convex solution spaces	Marler and Arora [17]
Weighted exponential sum	$\sum_{i=1}^{k} w_i [f_i(\vec{x})]^p$	Pareto optimality in non-convex solution spaces	Marler and Arora [17]
Weighted min–max	$MAX_{i=1}^{k}\{w_i [f_i(\vec{x}) - f_i^o]\}$	Weak Pareto optimality	Marler and Arora [17]
Weighted product	$\prod_{i=1}^{k} [f_i(\vec{x})]^{w_i}$	Different magnitudes in objectives	Marler and Arora [17]

3.1 How to Achieve Different Pareto Optimal Solutions

With the global objective sum in (2), just one solution to the optimization problem is obtained. However in a multi-objective optimization problem many optimal solutions exist, therefore to get different optimal solutions it is necessary to find the minima in different regions of the Pareto front. To reach this goal the following weighted AOF is used.

$$Z = \sum_{i=1}^{k} w_i f_i(\vec{x}) \tag{3}$$

If all the weights in (3) are positive, the minimum is Pareto optimal [16]. Diverse weights may achieve different Pareto optimal solutions; this is the core of how the multi-objective decomposition approach works. The original optimization with aggregate objective functions (AOFs) was developed using mathematical programming and different AOFs have been proposed to overcome various difficulties. In linear programming problems with convex solution spaces the use of weighted sum method guarantees an optimal solution, however for other problems is not possible to obtain points in the non-convex portions of the Pareto optimal set [17]. Table 1 shows a summary of weighted AOFs used in multi-objective optimization.

In mathematical programming, it is mandatory to select the correct optimization method with the correct AOF to guide the search according to the feasible decision space. It is important to notice that not all the weighted AOFs ensure Pareto optimality as in the weighted min–max (also known as weighted Tchebycheff), which only ensures weak Pareto optimality (none solution in the feasible space is better for all objectives). On the other hand with metaheuristics the search is guided by different operators, for example differential evolution is a great idea for convex problems to exploit linear dependencies, or polynomial mutations to handle non-convex problems [18]. Hybridization of heuristic operators and weighted AOF is a powerful tool for a more generic multi-objective optimization approach, although it does not guarantee optimality.

3.2 Decomposition of a Multi-objective Optimization Problem

In previous section, we reviewed how to achieve different Pareto optimal solutions, the weighted AOFs, to get Pareto optimal solutions. Now it is necessary to decompose the original multi-objective optimization problem into multiple single-objective problems in order to achieve a good representation of the Pareto Front. Let $W = \{\vec{w}_1, \vec{w}_2, \ldots, \vec{w}_n\}$, where $\vec{w}_i \leftarrow [w_{i1}, w_{i2}, \ldots, w_{ik}]$ is a weight vector. Most authors suggest $\sum_{j=1}^{k} w_{ij} = 1$, there is no real reason for this, but in order to achieve uniformity between solutions found it is necessary that the sum be the same for all the weight vectors defined. The selection of weight vectors is not trivial, similar weights bias the search towards the same region and very diverse weights will not produce uniformity. The first uses of the decomposition method suggest systematically altering weights to yield a set of Pareto optimal solutions [19, 17]. Currently the state of the art manages three main ideas to generate the weight vectors: random vector generators, systematic vector generators and optimization vector generators.

Random vector generators are the common approach. In the work of Zhang et al. [20] a vectors generator of this kind is used with good results. The generator initiates with the corners of the objective space (the best solution for every objective when the others are with 0 weight of importance) in the initial set W, an example of a corner is given in (4).

$$\vec{w}_1 \leftarrow [1.0, 0.0, 0.0, \ldots, 0.0] \tag{4}$$

Then the vector with the biggest distance to W is added from a set of 5,000 vectors uniformly randomly generated. The process is repeated until the desired size of W is reached.

Systematic vector generators use patterns to generate the weight vectors. In Messac and Mattson [21], the proposed systematic vector generator, weights are generated by increments of size $1/(n-1)$ such that the sum of weights is equal to one. The permutations formed with the weights of size k that satisfy $\sum_{j=1}^{k} w_{ij} = 1$ are used as in Fig. 1 from [21].

It is important to notice that when $k = n$ and the objectives have the same magnitude these weight vectors are similar to different orders in the lexicographic method. Another work with a systematic vector generator is in [22], it uses the Simplex-lattice design principle, where the components are formed with the equation $x_i = 0, \frac{1}{n}, \frac{2}{n}, \ldots, \frac{n}{n}$, the permutations formed with the components are used as weight vectors to achieve a uniform Pareto front.

It is possible to formulate a vectors generator as an optimization problem. The work of Hughes [23] proposed generate the weight vectors using the following optimization problem.

$$Minimize\ Z = MAX_{i=1}^{n}\left\{MAX_{j=1, j\neq i}^{n}\left(\vec{w}_i \cdot \vec{w}_j\right)\right\} \tag{5}$$

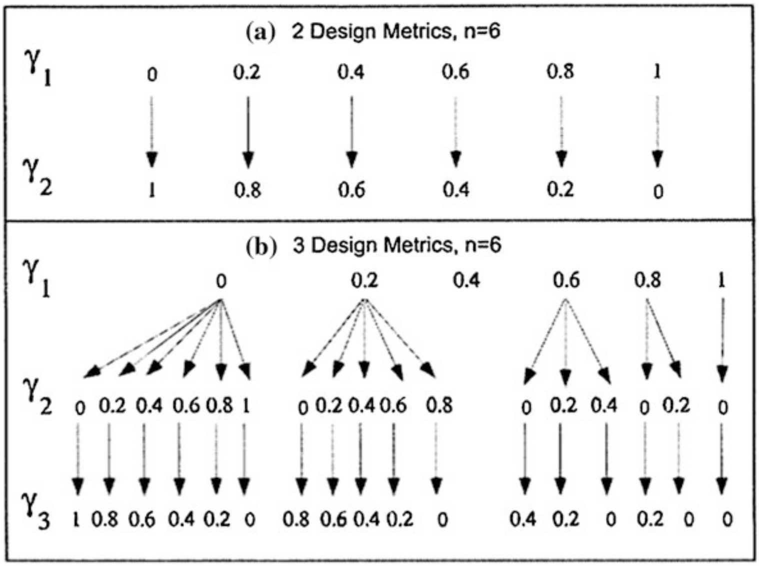

Fig. 1 Permutations of weights

The inner maximum operator finds the nearest vectors in angle, because the dot product provides the cosine of the angle between $\vec{w}_i$ and $\vec{w}_j$ and the minimum angle provides the maximum value $\cos(0) = 1$. The outer maximization operator finds the smallest of the angles in the vector set. It is important to clarify this vectors generator works with one assumption: the real optimal solution is when all the vectors in W are the same. In order to avoid this condition, all the corners of the objective space have to be added to the set W and remain in the entire optimization process.

3.3 General Multi-objective Optimization Framework by Decomposition

The multi-objective decomposition algorithms are based on population search algorithms (evolutionary algorithms, ant colony, and particle swarm) or path search. Figure 2 shows the framework used by most of the multi-objective decomposition algorithms.

The framework requires every vector $\vec{w}_i$ be in a cluster of similar vectors from the set W. The framework works in four steps. The first step sets an initial solution $\vec{x}_i$ for every vector $\vec{w}_i$.The second step selects one $\vec{w}_i$ from the set W and optimizes it (with any optimization method) from the best-known solution $\vec{x}_i$ for that vector. The third step uses the new solution found $\vec{x}'$ to update the best-known solutions for the weight vectors clustered with $\vec{w}_i$. The fourth step verifies if the stop criteria

Require: Every vector $\vec{w}_i$ be in a cluster C_z
for $\forall \vec{w}_i \in W$ **do**
  $\vec{x}_i \leftarrow$ an initial solution for the vector $\vec{w}_i$
end for
repeat
  select one vector $\vec{w}_i \in W$
  $\vec{x}' \leftarrow Optimize(\vec{x}_i)$;
  for $\forall \vec{w}_j \in C_z$ **do** ▷ C_z are the clustered vectors to $\vec{w}_i$
    if $\vec{x}' < \vec{x}_j$ **then**
      $\vec{x}_j \leftarrow \vec{x}'$
    end if
  end for
until stop criterion

Fig. 2 Multi-objective optimization framework by decomposition

have been reached; if not, return to the second step. Overlapping in the clusters is also allowed. For example, sometimes the optimization of a vector updates the entire population or sometimes a smaller portion. In the first multi-objective decomposition proposals of Hughes, the structure is different from Fig. 2, MSOPS [24] and MSOPS-II [23]. Later Hughes changed to the structure of Fig. 2 in his last decomposition approach MODELS [25] with better performance results.

3.4 Multi-objective Optimization Based on Decomposition Versus Dominance Based

The multi-objective decomposition methods have benefits over based dominance optimization methods. With dominance and relaxed dominance forms the non-dominance verification is a very repetitive operation of order $O(kn)$, where n is the set of non-dominated solutions and k the number of objective functions. While in decomposition methods the comparisons are directly between solutions in $O(k)$. When the number of objectives increases, the number of non-dominated solutions grows [26], which leads to stagnation in the search processes based on dominance, because for the optimizer most solutions are equally good.

4 Current Research on Decomposition Methods

This section reviews the current research trends on multi-objective decomposition methods: heuristic approaches, performance, efficiency, and scalability.

The current heuristic approaches are based on well-known metaheuristic optimization algorithms methods: differential evolution, memetic algorithms, ant colony and particle swarm optimization.

Differential evolution is an approach that emphasizes mutation using a proportional difference of two random vector solutions to mutate a target vector.

In multi-objective decomposition the first implementation found is from Li and Zhang [27], an evolution of the first MOEA/D in Zhang and Li [14], called MOEA/D-DE. The original version of MOEA/D keeps an external archive with all the non-dominated solutions found. MOEA/D-DE does not use an external archive because the final solutions for the set W are usually non-dominated. Another modification is a parameter of maximal number of solutions updated by a new one found. This algorithm has been tested with very hard Pareto shapes.

Memetic algorithms are based on the synergy of 2 ideas, population search and local improvements a very robust idea for optimization. MOGLS in Ishibuchi and Murata [28] was the first memetic proposal in multi-objective decomposition, but later an evolutionary algorithm MOEA/D in Zhang and Li [14] obtains a very similar performance without local improvements. In the work of Tan et al. [29] MOEA/D-SQA, the population search is the MOEA/D, and the local improvements are made by the simplified quadratic approximation (SQA), outperforming MOEA/D. Another memetic algorithm is in Mei et al. [30] for the capacitated arc routing problem (CARP), where the population search is a modification of the MOEA/D structure. The solutions are preserved using the preservation mechanism from NSGA-II [31], and the preserved solutions are assigned to the weight vector that solves best. The crossover and local search operators are from the memetic algorithm with extended neighborhood search (MAENS).

Ant colony optimization emulates the behavior of ants looking for a path between food and their colony; ants with promissory paths leave a pheromone path to be followed for other ants. The ants (solutions) are constructed stochastically between the best path and others. This optimization method uses 2 kinds of information, pheromone and heuristic. The pheromones represent the paths leading to promissory solutions, while heuristic information is how the solutions are constructed (for example restrictions). The work of Ke et al. [32] proposed MOEA/D-ACO where each ant is a weight vector to optimize and has its own heuristic information. The pheromone information is shared with a group of solutions (not necessarily the clustered weight vectors) this group tries to achieve a region of the Pareto front. Every ant is constructed with its heuristic information, pheromone path, and best local solution known; this information is combined to guide the ant.

Particle Swarm optimization (PSO) is inspired by the social foraging behavior of some animals, such as flocking behavior of birds and the schooling behavior of fish. Velocity guides the search of the particles, fluctuating between their own best past location (best local) and the best past location of the whole swarm (best global/leader). Peng and Zhang [33] proposed the first PSO based on multi-objective decomposition MOPSO/D. Every particle is assigned to a weight vector, MOPSO/D uses a crowding archive strategy to keep the non-dominated solutions found. Later Al Moubayed et al. [34] propose SDMOPSO, the main difference with MOPSO/D is that it uses Pareto dominance to update the best locations local and global; the leaders are selected from a random non-dominated particle from its clustered vectors. Martínez and Coello [35] proposed dMOPSO, which does not use turbulence (mutation) and the leader is a random solution from the best local particles in the swarm. In dMOPSO the concept of age is introduced, if a particle is not improved in a generation its age is increased by one; the particles that exceed certain age are reinitialized from scratch. In the work of Al Moubayed et al. [36] D^2MOPSO was proposed comparing it with SDMOPSO and dMOPSO, outperforming them. D^2MOPSO uses a crowding archive, when the archive is full the non-dominated particles are only added at the low dense regions replacing those ones at high dense regions. The selections of leaders are made from the archive using the solution that best solves the weight vector of the particle.

The performance is measured by the quality in the Pareto front representation found, a very competitive field in multi-objective optimization. There are different desirable characteristics: convergence, uniformity and a big dominated region. The inverted generational distance [37] measures the distance from the real front to the representation. It was the indicator used in the IEEE CEC 09 competition, where the winner for unconstrained problems was MOEA/D [20]; this version of MOEA/D uses a utility function that assigns less computational effort when the weight vectors are not improved in the generation.

Efficiency is an issue hard to improve in multi-objective optimization, the reason for this is the computational efficiency of non-dominance verification and sorting in fronts $O(kn^2)$ [38, 31]. The multi-objective decomposition methods do not use dominance, allowing lower execution times in multi-objective optimization. Bearing in mind Zhang and Li design MOEA/D in [14] with a lower complexity in space and time and similar performance than MOGLS in [28].

The scalability in objectives is a hard topic in multi-objective optimization because optimization methods based on dominance have difficulties. It is well known that dominance based optimization methods lose their effectiveness with more than 3 objectives [39]. When the number of objectives grows it increases the number of non-dominated solutions [26], causing stagnation in the search. Hughes proposed an evolutionary algorithm MSOPS [24] and improved it in MSOPS-II [23], for a problem of 5 objectives outperforming NSGA-II. Later Hughes proposed a path search based on golden section search named MODELS in [25], for a problem of 20 objectives outperforming MSOPS-II. The work of Yan et al. [22] UMOEAD ensures a uniform Pareto front, using a systematic vectors generator, tested with a 5 objectives problem.

5 Future Research on Decomposition Methods

This section presents the authors perspective about what is still need to be researched in multi-objective decomposition methods.

5.1 Weight Vector Generators

In order to achieve a good representation of the Pareto front, the final solutions have to be diverse enough to cover most of the Pareto front and the extremes; also uniformity is important to not ignore regions in the Pareto front. The problem with the current vector generators is that they only focus in diversity or uniformity, but not both. Random vector generators are just focused in diversity while systematic vector generators are just targeting uniformity. Another issue is that when increasing the number of objectives usually the current vector generators use more vectors to ensure diversity or uniformity, implying more computational effort in the optimization. The multi-objective decomposition needs new vector generators not sensitive to the number of vectors taking account the diversity and the uniformity.

5.2 Problem Clustering and Communications

The current multi-objective decomposition methods gather the problems with the metric of Euclidian distance and use the same cluster technique, the T closest problems. It is possible to apply other clustering techniques like k-means or k-medoids, and metrics like Manhattan distance or Mahalanobis distance [40], to the multi-objective decomposition methods. There are no studies about different metrics or cluster techniques and the general structure of these optimization methods is affected directly by the clusters of weight vectors. Different clustering techniques and metrics could be used and studied. Also the communication between clusters of vectors is generally random, a more structured communication mechanism could be designed to exploit clusters information and improve another ones.

5.3 New Heuristic Approaches

The current multi-objective decomposition methods are based on metaheuristic population search or path search, there is a great variety of population search algorithms not studied yet and path search is almost not studied. A search based on clusters and its communications could be a new topic for heuristic approaches.

5.4 Scalability in Decision Variables

Although scalability in objectives is studied by some multi-objective decomposition methods, the scalability in the number of decision variables is barely studied in multi-objective optimization; this was studied in the work of Durillo et al. [41].

The scalability in decision variables is not studied yet in multi-objective decomposition. It is important to notice that when some problems grow in objectives it also grows in decision variables [42, 43].

6 Conclusion

The multi-objective decomposition methods solve problems related to methods based on dominance forms like stagnation for non-dominated solutions and the need of dominance verification $O(kn)$ or sorting $O(kn^2)$ [31]. More research is needed on multi-objective decomposition methods in: vector generators, clustering techniques and heuristic approaches. Research on multi-objective decomposition may lead to diverse and uniform Pareto front representations for a wide variety of problems.

Acknowledgments B. Dorronsoro acknowledges the support by the National Research Fund, Luxembourg (AFR contract no. 4017742). A. Santiago would like to thank CONACyT Mexico, for the support no. 360199.

References

1. Dantzig, G.B., Wolfe, P.: Decomposition principle for linear programs. Oper. Res. **8**(1), 101–111 (1960)
2. Dantzig, G.B., Harvey, R.P., Lansdowne, Zachary F., Robinson, D.W., Maier, S.F.: Formulating and solving the network design problem by decomposition. Transp. Res. Part B: Methodol. **13**(1), 5–17 (1979)
3. Sobieszczanski-Sobieski, J., James, B.B., Dovi, A.R.: Structural optimization by multi level decomposition. AIAA J. **23**(11), 1775–1782 (1985)
4. Ovacik, I.M., Uzsoy, R.: Decomposition Methods for Complex Factory Scheduling Problems. Kluwer Academic Publishers, Boston (1997)
5. Dorronsoro, B., Danoy, G., Nebro, A.J., Bouvry, P.: Achieving super-linear performance in parallel multi-objective evolutionary algorithms by means of cooperative coevolution. Comput. Oper. Res. **40**(6), 1552–1563(2013) (Emergent nature inspired algorithms for multi-objective optimization)
6. Liu, M., Zou, X., Chen, Y., Wu, Z.: Performance assessment of DMOEA-DD with CEC 2009 MOEA competition test instances. In: IEEE Congress on Evolutionary Computation (CEC 09), pp. 2913–2918 (2009)
7. Boyd, S., Xiao, L., Mutapcic, A., Mattingley, J.: Notes on decomposition methods. Notes for EE364B, Stanford University (2008)
8. Abdelouahed, H., Mishra, S.K.: Decomposition methods based on augmented lagrangians: a survey. In: Mishra, S.K. (ed.) Topics in Nonconvex Optimization, Springer Optimization and Its Applications, pp. 175–203. Springer, New York (2011)
9. Koko, J.: A survey on dual decomposition methods. SeMA J. **62**(1), 27–59 (2013)
10. Palomar, D.P., Chiang, Mung: A tutorial on decomposition methods for network utility maximization. IEEE J. Sel. Areas Commun. **24**(8), 1439–1451 (2006)

11. Sakawa, M., Yano, H.: A fuzzy decomposition method by right-hand-side allocation for large-scale multiobjective nonlinear programming problems. In: Clmaco, J. (ed.) Multicriteria Analysis, pp. 237–246. Springer, Berlin, Heidelberg (1997)
12. Sefrioui, M., Perlaux, J.: Nash genetic algorithms: examples and applications. In: Proceedings of the 2000 Congress on Evolutionary Computation, vol. 1, pp. 509–516 (2000)
13. Zhan, Z.H., Li, J., Cao, J., Zhang, J., Chung, H.S.-H., Shi, Y.-H.: Multiple populations for multiple objectives: a coevolutionary technique for solving multiobjective optimization problems. IEEE Trans. Cybern. **43**(2), 445–463 (2013)
14. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007)
15. Grabisch, M., Marichal, J.-L., Mesiar, R., Pap, E.: Aggregation functions: means. Inf. Sci. **181**(1), 1–22 (2011)
16. Zadeh, L.: Optimality and non-scalar-valued performance criteria. IEEE Trans. Autom. Control **8**(1), 59–60 (1963)
17. Marler, R.T., Arora, J.S.: Survey of multi-objective optimization methods for engineering. Struct. Multi. Optim. **26**(6), 369–395 (2004)
18. Sindhya, K., Ruuska, S., Haanp, T., Miettinen, K.: A new hybrid mutation operator for multiobjective optimization with differential evolution. Soft. Comput. **15**(10), 2041–2055 (2011)
19. Athan, T.W., Pa-palambros, P.Y.: A note on weighted criteria methods for compromise solutions in multi-objective optimization. Eng. Optim. **27**, 155–176 (1996)
20. Zhang, Q., Liu, W., Li, H.: The performance of a new version of MOEA/D on CEC09 unconstrained mop test instances. In: IEEE Evolutionary Computation (CEC'09), pp. 203–208 (2009)
21. Messac, A., Mattson, C.A.: Generating well-distributed sets of Pareto points for engineering design using physical programming. Optim. Eng. **3**(4), 431–450 (2002)
22. Tan, Y.Y., Jiao, Y.C., Li, H., Wang, X.: MOEA/D uniform design: a new version of MOEA/D for optimization problems with many objectives. Comput. Oper. Res. **40**(6), 1648−1660 (2013) (emergent nature inspired algorithms for multi-objective optimization)
23. Hughes, E.J.: Msops-ii: a general-purpose many-objective optimiser. In: IEEE Congress on Evolutionary Computation (CEC 2007), pp. 3944–3951 (2007)
24. Hughes, E.J.: Multiple single objective Pareto sampling. In: The 2003 Congress on Evolutionary Computation (CEC'03), vol. 4, pp. 2678–2684 (2003)
25. Hughes, E.J.: Many-objective directed evolutionary line search. In: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO'11), pp. 761–768. ACM, New York (2011)
26. Fabre, M.G., Pulido, G.T., Coello, C.A.C.: Alternative fitness assignment methods for many-objective optimization problems. In: Collet, P., Monmarch, N., Legrand, P., Schoenauer, M., Lutton, E. (eds.): Artificial Evolution. Lecture Notes in Computer Science, vol. 5975, pp. 146–157. Springer, Berlin, Heidelberg (2010)
27. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. IEEE Trans. Evol. Comput. **13**(2), 229–242 (2009)
28. Ishibuchi, H., Murata, T.: A multi-objective genetic local search algorithm and its application to flowshop scheduling. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **28**(3), 392–403 (1998)
29. Tan, Y.-Y., Jiao, Y.-C., Li, H., Wang, X.-K.: MOEA/D-SQA: a multi-objective memetic algorithm based on decomposition. Eng. Opt. **44**(9), 1095–1115 (2012)
30. Mei, Y., Tang, K., Yao, X.: Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem. IEEE Trans. Evol. Comput. **15**(2), 151–165 (2011)
31. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In: Proceedings of the 6th International Conference on Parallel Problem Solving from Nature. Lecture Notes in Computer Science, vol. 1917. Springer, Berlin, Heidelberg (2000)

32. Ke, L., Zhang, Q., Battiti, R.: MOEA/D-ACO: a multiobjective evolutionary algorithm using decomposition and ant colony. IEEE Trans. Cybern. **43**(6), 1845–1859 (2013)
33. Peng, W., Zhang, Q.: A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems. In: IEEE International Conference on Granular Computing (GrC 2008), pp. 534–537 (2008)
34. Al Moubayed, N., Petrovski, A., McCall, J.: A novel smart multi-objective particle swarm optimisation using decomposition. In: Schaefer, R., Cotta, C., Koodziej, J., Rudolph, G. (eds.) Parallel Problem Solving from Nature, PPSN XI. Lecture Notes in Computer Science, vol. 6239, pp. 1–10. Springer, Berlin Heidelberg (2010)
35. Martínez, S.Z., Coello, C.A.C.: A multi-objective particle swarm optimizer based on decomposition. In: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO'11), pp. 69–76. ACM, New York (2011)
36. Al Moubayed, N, Petrovski, A., McCall, J.: D^2MOPSO: multi-objective particle swarm optimizer based on decomposition and dominance. In: Hao, J.-K., Middendorf, M. (eds.) Evolutionary Computation in Combinatorial Optimization. Lecture Notes in Computer Science, vol. 7245, pp. 75–86. Springer Berlin Heidelberg (2012)
37. Van Veldhuizen, D.A., Lamont, G.B.: Multiobjective Evolutionary Algorithm Research: A History and Analysis (1998)
38. Coello, C.A.C.: Evolutionary multi-objective optimization: some current research trends and topics that remain to be explored. Front. Comput. Sci. China **3**(1), 18–30 (2009)
39. Hughes, E.: Evolutionary many-objective optimisation: many once or one many? In: The 2005 IEEE Congress on Evolutionary Computation, 2005, vol. 1, pp. 222–227 (2005)
40. Berkhin, P.: A survey of clustering data mining techniques. In: Kogan, J., Nicholas, C., Teboulle, M. (eds.) Grouping Multi-dimensional Data, pp. 25–71. Springer, Berlin, Heidelberg (2006)
41. Durillo, J.J., Nebro, A.J., Coello, C.A.C., Garcia-Nieto, J., Luna, F., Alba, E.: A study of multi-objective metaheuristics when solving parameter scalable problems. IEEE Trans. Evol. Comput. **14**(4), 618–635 (2010)
42. Laumanns, M., Deb, K., Thiele, L., Zitzler, E.: Evolutionary Multiobjective Optimization. Theoretical Advances and Applications. Scalable Test Problems for Evolutionary Multi-Objective Optimization. 105−145 (2005)
43. Barone, L., Huband, S., Hingston, P., While, L.: A review of multi-objective test problems and a scalable test problem toolkit. IEEE Trans. Evol. Comput. **10**(5), 447–506 (2006)