

Balanced Explore-Exploit Clustering based Distributed Evolutionary Algorithm for Multi-objective Optimisation

Mariem Gzara¹, Abdelbasset Essabri²

¹ Multimedia Information systems and Advanced Computing Laboratory (MIRACL),
route Tunis Km 3; B.P. 1030, Sfax, 3018
mariem.gzara@isimsf.rnu.tn

² Laboratoire de Gestion Industrielle et d'Aide à la Décision (GIAD),
route de l'aérodrome Km 4.5 BP 1088, Sfax, 3018
abdelbasset.essabri@fsegs.rnu.tn

Abstract: Most parallel evolutionary algorithms for single and multi-objective optimisation are motivated by the reduction of the computation time and the resolution of larger problems. Another promising alternative is to create new distributed schemes that improve the behaviour of the search process of such algorithms. In multi-objective optimisation problems, more exploration of the search space is required to obtain the whole or the best approximation of the Pareto Optimal Front. In this paper, we present a new clustering-based parallel multi-objective evolutionary algorithm that balances between the two main concepts in metaheuristics, which are exploration and exploitation of the search space. The proposed algorithm is implemented and tested on several standard multi-objective test functions using a network of multiple computers.

Keywords: parallel computing, multi-objective optimisation, evolutionary algorithms.

1. Introduction

Evolutionary Algorithms (EAs) are stochastic search metaheuristics that have been used successfully to solve many optimisation problems such as scheduling, routing, etc [2, 17, 23, 14, 15]. EAs are population based metaheuristics that perform well global search by exploring simultaneously different regions of the search space. They are therefore less attracted to local optima and are suitable to solve Multi-Objective Optimisation problems (MOP) where a set of non-dominated solutions is sought. This is in contrast to the single-objective optimisation problems where a unique optimum is sought. Several algorithms are proposed to adapt EAs to MOPs, including NSGA-II [9, 25], SPEA-II [28], NPGA-II [20]. These algorithms can be divided into two categories: the non-Pareto and the Pareto Multi-Objective Evolutionary Algorithms (MOEAs). The concept of Pareto dominance is used to rank the population in such a way that all non-dominated individuals in the population are assigned the same cost. Pareto elitist MOEAs maintain another population than the current one, which permits to keep the Pareto-optimal solutions found during the search. This external population participates in the process of selection. Thus, elitism permits a better intensification of the search. Among these algorithms, SPEA-II [28] and NSGA-II [25] have powerful search mechanisms and

obtained good results. SPEA-II [28] and NSGA-II [25] use similar concepts; such as Pareto selection, elitism and diversification techniques that are proved to be efficient to characterise the Pareto Optimal Front (POF). A state of the art on MOEAs is given in [13, 8, 7].

Despite their success in solving several real-world MOPs [11, 16], these algorithms require large computational times and memory. Several parallel schemes are proposed to overcome this problem. Another motivation is to exploit parallelism to best explore the search space and to discover new ways to direct the search. In the single-objective case, parallel EAs exploit the inherent parallelism in evolutionary computation where crossover, mutation, selection and fitness evaluation can be easily distributed. Whereas, in the multi-objective case, many parallel schemes focus on how to divide the population on the objective or/and the decision space where each process will concentrate on a specific region of the search space.

One of the most difficult issues in designing metaheuristics for the resolution of both single-objective and multi-objective problems is how to balance between exploration and exploitation of the search space. For example, in evolutionary computation a high rate of conventional mutation (high diversification) makes the search process looks like a random exploration. While a high rate of crossover

without mutation leads to a premature convergence. In Parallel Multi-Objective Evolutionary Algorithms (PMOEAs), the parallelism can be exploited to direct the search toward more exploration or more exploitation through the mechanisms of collecting, dividing and redistributing the global population or a subset of individuals among the available processors. In this paper, we propose a new parallel evolutionary algorithm for multi-objective optimisation that is called “Balanced Explore Exploit clustering based Distributed Evolutionary Algorithm for multi-objective optimisation” (BEEDEA). This algorithm is based on dividing the search space by clustering algorithms then redistributing the individuals such that both global search and local search will be performed. To test the BEEDEA, we use the benchmark functions of Zitzler [27] on a network of several computers. Experimental results have shown that the BEEDEA performs a good balance between exploration and exploitation of the search space and it is more efficient to characterise the POF than a classic island model parallel MOEA without migration.

The paper is organised as follows. Section 2 reviews the literature on existing PMOEAs. Section 3 describes the proposed parallel MOEA. Section 4 presents numerical results. Section 5 concludes the paper and outlines future research directions.

2. Parallel Multi-objective Evolutionary Algorithms

Parallel EAs are classified into three categories: *master-slave population model*, *island model*, and *cellular model* [5, 1]. The master-slave model uses a simple distribution of the fitness calculation stage. In the island model, the entire population is divided into small groups which are called sub-populations or islands. Each sub-population evolves on a separate processor, following a traditional diagram. Generally, a migration stage is added. In the cellular model, one or more individuals are set to evolve on each processor. Then the selection, replacement and crossover operations are performed between individuals that are neighbours on the topology of the processor network.

Several proposed PMOEAs are based on the specialisation concept [3, 18, 26] which means dividing the objective and/or the search space then assigning each part to a processor. One processor called the organiser or the coordinator is usually charged to direct the whole algorithm.

In Branke et al. [3], the cone separation technique is based on normalising the fitness values and partitioning the fitness space into cones by starting from the reference point (1,1), dividing the 90° angle encompassing the non-dominated front into equal parts. Each processor will focus on one region since the borders of each region are treated as constraints. At regular intervals, renormalisation of the fitness space and re-adaptation of the constraints causes individuals violating the constraints of a sub-population to migrate into the sub-population where they do not violate the constraints. This approach has several drawbacks. If the PF is discontinued or is not evenly distributed, one processor may receive an empty or small sub-population (for example extreme points in two discontinuous parts of the front). When the number of processors increases, the cone (region) allocated to each one becomes small, and the number of offspring’s violating the constraints increases. This leads to less diversified sub-populations, and so large communication time and slow convergence. Moreover, this subdivision technique becomes rather complicated in the case of more than two objectives.

In the approach of Hiroyasu [18, 19], the population is gathered at regular intervals, sorted according to one of the objectives (objective used for sorting is chosen in turn) and divided into sub-populations. A simple MOEA is performed for each sub-population. In the case of discrete or discontinuous POF, this approach has generated Pareto solutions which are concentrated on some points rather than all over the POF. In fact the partition is somewhat arbitrary according to changing objectives and does not guide the search toward the less explored regions. In [6] De Toro et al. have presented a similar parallel scheme as [18] with the difference of using a Single Front Genetic Algorithm on each processor. In the Multi-Objective Genetic Algorithm with Distributed Environment Scheme (MOGADES) proposed by Kamiura

et al. [22], the multi-objective problem is transformed into a single objective problem which is solved in each island using different weight parameters. Setting the weights correctly requires a certain amount of search space knowledge, which is often not available in advance. Moreover, this parallel MOEA could miss some Pareto optimal solutions when the search space is non-convex. Such solutions are called not supported. So, this way of dealing with multi-objective optimisation is not always possible or efficient.

In Streichert et al. [26], the sub-populations are gathered, clustered by the k-means clustering algorithm (the number of clusters is equal to the number of processors) and then redistributed onto the available processors. The constrained dominance principle [4] using the cluster centroid (each individual is assigned to one cluster centroid) is implemented to limit sub-populations to their specific regions. The drawbacks of this approach are that the generated individuals outside the zone constraints are not considered and this may limit the diversity of sub-populations. Furthermore, the clustering algorithm divides the search space into a number of regions which is equal to the number of processors. However, this number depends on the distribution of individuals in the search space.

Deb et. al. [10] use the migration and the guided dominance principle [4] to give the different islands different search directions. The drawback of the approach is that it may be difficult to define appropriate search directions before the shape of the POF is known. Moreover this approach can be applied only when the POF is convex.

In [12], Essabri et al. presented the parallel multi-Objective evolutionary algorithm with Multi-Front Equitable Distribution (MFED). At regular intervals, MFED gathers the k first global Pareto fronts (k is a parameter) constructed from all processors then distributes them with equity between all sub-populations. The i^{th} ($i \in \{1..k\}$) global Pareto front is at first partitioned into NC clusters (NC is a parameter) using an agglomerative clustering algorithm. Then each cluster is distributed on the processors. Each sub-population receives at least one individual from each cluster (Figure 1). The main drawback of MFED is that neighbouring solutions are distributed to distinct clusters. The MFED doesn't perform well neighbourhood searches and converges slowly.

In this paper, we propose a new parallel scheme for MOEAs. The BEEDEA balances between the two main concepts in metaheuristics which are local and global exploration.

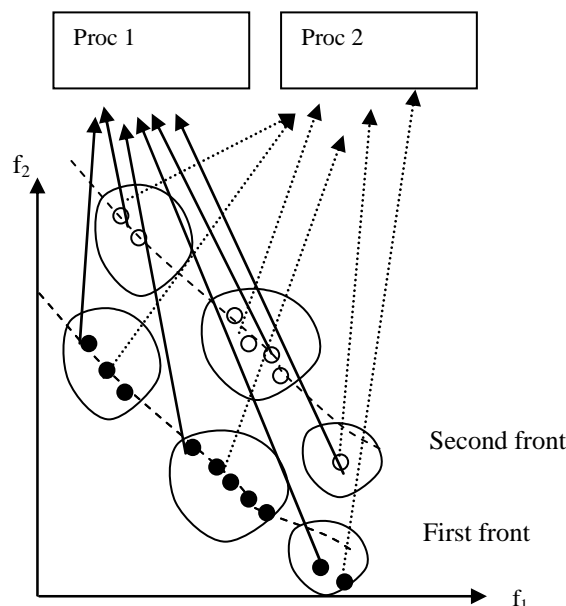


Figure 1. The parallel multi-Objective evolutionary algorithm with Multi-Front Equitable Distribution

3. Balanced Explore-Exploit Clustering based Distributed Evolutionary Algorithm for Multi-objective Optimisation (BEEDEA)

There are two tasks that a MOEA should accomplish in solving MOPs: guide the search towards the global Pareto-optimal region, and maintain population diversity (in the objective space, search space, or both) in the current non-dominated front. Since MOEAs search for a whole set of diversified solutions (the Pareto optimal front) instead of a single solution, many parallel multi-objective evolutionary schemes [3, 26, 18] are based on dividing the optimisation problem (the search space and/or the objective space) into multiple sub-problems such that each process will focus on one region of the search space or one part of the Pareto front. These parallel schemes are shown to be more efficient to solve MOPs than sequential MOEAs. The question is how to find a suitable partitioning of a given optimisation problem and how to distribute the individuals on the available processors?

research toward the unvisited regions. It permits to generate solutions that differ meaningfully of those previously visited. The algorithm executes iteratively and in turn an exploration step and exploitation step (Figure 2). In the exploitation step, solutions which are close to each other are collected into one sub-population. A clustering algorithm divides the global population into p clusters (p is the number of processors) so that individuals in the same cluster are similar between themselves and dissimilar to individuals of other groups. Each processor receives one cluster (one sub-population), which will evolve for few generations. By this way, intensification is performed by favouring the recombination of neighbouring solutions.

In the exploration step, solutions belonging to different zones of the search space are grouped together into one sub-population. The global population is partitioned into $2p$ clusters by a clustering algorithm. Each processor receives 2 distinct clusters. Diversification is performed while favouring the recombination of distant individuals. In fact, a crossover stage between pairs of individuals each one of them from one cluster

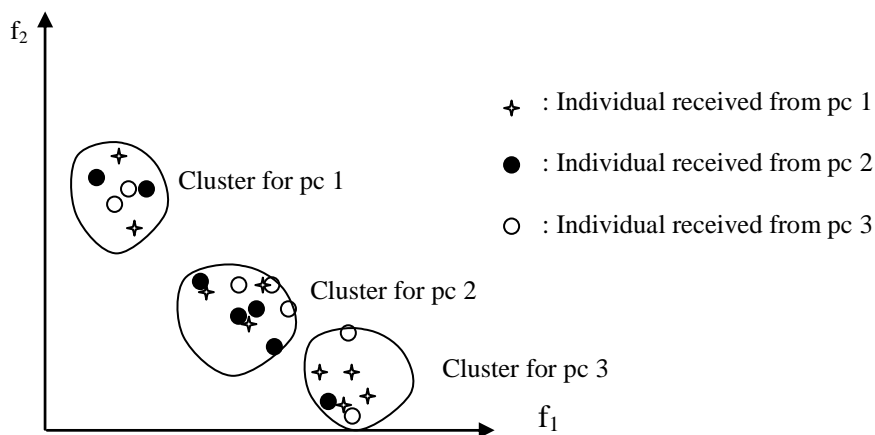


Figure 2. Clustering for the exploitation step with 3 processors.

The basic idea of the proposed parallel scheme is motivated by the fact that the efficiency of the search process is related to obtaining a good balance between the two main search mechanisms: exploitation and exploration. The exploitation permits to explore regions of the search space near of those visited earlier that have good quality. However, the exploration tries to direct

is performed. After that a MOEA will be performed for few generations.

In our approach, a hierarchical agglomerative clustering algorithm is used to partition the global population of N individuals into sub-populations. These individuals are considered in the objective space. The agglomerative clustering method starts by regarding these as

N separate clusters of size 1. Then, finds the most similar pair of clusters and merges them together into a single cluster to reduce the number of clusters to $(N-1)$. This procedure continues until the desired number of clusters is reached. To compute the similarity between two individuals, we use the Euclidian distance in the fitness space. We consider the distance between one cluster and another cluster to be equal to the Euclidian distance between their respective centers of gravity. Since the fitness space to be searched and the shape of the Pareto-Optimal Front (POF) are usually unknown at the start of the optimisation; such a partitioning of the fitness space has to be adapted at regular intervals.

The main algorithm consists of two kinds of processes, an organizer and several elitist MOEAs. At regular intervals, the organizer collects the different sub-populations from the other processors, divides the global population into p clusters in the exploitation step and $2p$ clusters in the exploration step. Then it distributes them among all processors (the organizer and the others). Every sub-population will evolve with its proper genetic operators (crossover and mutation) to explore differently the search space. Moreover, the use of different genetic environment reduces the risk of performing the same individuals in the different islands. The model can be explained as follow:

Interval $\leftarrow 0$

Step 1: Initialise sub-population is performed randomly. The global population size is N . The size of each sub-population is N/p (p is the processor number).

Step 2: Multi-objective genetic algorithm is performed for some generations. The MOEA used in this paper and the genetic operators are given in the next section.

Step 3: All the individuals are gathered. Then the global PF is formed.

Interval \leftarrow Interval +1.

If Interval mod 2=0 then go to step 4 else go to step 5.

Step 4:

- A clustering algorithm is performed on the global population. The number of clusters is equal to p .
- Exploitation step: send each cluster to one processor. Each processor will execute a MOEA for some iterations. Go to step 3.

Step 5:

- A clustering algorithm is performed on the global population. The number of clusters is equal to $2p$.
- Exploration step: each process receives two distinct clusters and a crossover step between pairs of individuals each one from one cluster is performed and then a MOEA runs for some generations. Go to step 3.

4. Experimental Results

Our algorithm is implemented using C++ language. We ran our algorithm on a cluster using several processors. The communication between the processors is supported by Message Passing Interface library (MPI). Our algorithm is based on an elitist MOEA, the Strength Pareto Evolutionary Algorithm (SPEA) [29], which runs on each processor. SPEA keeps individuals whose decision vectors are non-dominated in an external population. When this set exceeds the maximum archive size, it reduces the number of individuals externally sorted by means of clustering. In order to set the parameters of this algorithm, several tests have been realised with different occurrences. These parameters are fixed as follows:

- Population size (N): 200 individuals between the available processors;
- Size of the global archive: 200 individuals between the available processors;
- Number of generations (T): 2000;
- Crossover probability (P_c): 0.99;
- Mutation probability (P_m): 0.1;
- Crossover type: one point, two points and uniform crossover;
- Mutation type: bit inversion mutation;

The gathering, partitioning and redistribution of the global population are performed at regular intervals of 15 generations. We have tested our approach on the benchmark functions of Zitzler [27]: ZDT2 (concave Pareto optimal front) and ZDT6 (non-convex and non-uniformly distributed Pareto optimal front). Both test problems considered here are bi-objective. We compare the following parallel configurations:

- PSPEA: A classic island parallel MOEA without migration based on SPEA.
- BEEDEA: The Balanced Explore/Exploit clustering based Distributed Evolutionary Algorithm for Multi-objective Optimisation.
- Configuration D: The BEEDEA with only reiteration of the exploration step.
- Configuration I: The BEEDEA with only reiteration of the exploitation step.

The results are compared using the metrics of GUIMOO (Graphical User Interface for Multi-Objective Optimization [21] which offers tools to analyse results of multi-objective methods. To properly evaluate and compare MOEA performances, we identify three important criteria [7]:

- Accuracy: the distance of the resulting non-dominated set to the POF should be minimal.
- Uniformity: the solutions should be well distributed (in most cases uniform).
- Extent: the non-dominated solutions should cover a wide range for each objective function value.

For this reason, to compare and evaluate the non-dominated sets, we have used the S metric for accuracy, the spacing for the uniformity and some figures to see if the non-dominated set covers the totality of the Pareto front. We have added the entropy as a measure of diversity and the contribution to compare two Pareto Fronts (PF). Since evolutionary algorithms are stochastic search methods, each algorithm is run five times. Then, each metric is calculated as the mean of the five executions.

ZDT2 problem:

We apply our algorithm to the bi-objective and 30 variable ZDT2 problem. The Pareto front of that problem is concave. Table 1 gives the results for the BEEDEA and the PSPEA on 5 processors. We conclude that the non-dominated set generated by the BEEDEA is well distributed ($\text{spacing}(\text{BEEDEA}) < \text{spacing}(\text{PSPEA})$). But it is less diversified than that obtained by the PSPEA. The entropy values for BEEDEA and PSPEA are respectively 0.39 and 0.35. The results for the S metric are similar.

Table 1. Comparison of the BEEDEA and the PSPEA for the ZDT2 problem

	PSPEA	BEEDEA
Contribution	0,53	0,47
S Metric	0,3206914	0,320606
Entropie	0,39020272	0,35463096
Spacing	0,01842744	0,01100374

In the previous section, we have explained that the parallel scheme proposed in this paper performs a good balance between exploration and exploitation through the reiteration of the exploration step and the exploitation step. These two steps are essential and complementary for the efficiency of our approach. To validate this idea, we compare the performance of the algorithm when we omit one of them. Table 2 compares the performance when only exploration step (D) or exploitation step (I) is performed for the ZDT2 problem on 2, 3, 4 and 5 processors. Figure 3 shows the PFs generated by configurations I and D on 5 processors.

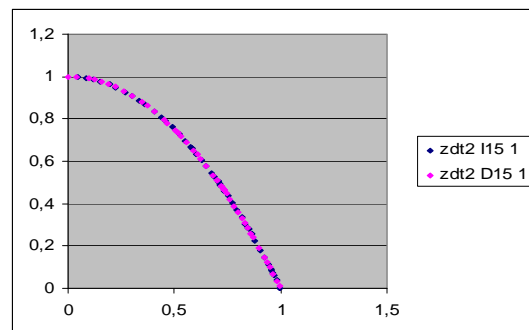


Figure 3. Non dominated solutions obtained for the ZDT2 problem with configurations I (zdt2 I) and D (zdt2 D) on 5 processors

Table 2. Performance of the non dominated solutions obtained for the ZDT2 problem with configurations I and D

Number of processors	Spacing		Contribution		S metric	
	D	I	D	I	D	I
2	0.018965	0.0156447	0.530612	0.469388	0.320898	0.320856
3	0.0168738	0.0157057	0.505495	0.494505	0.321028	0.320666
4	0.0179484	0.0174855	0.5	0.5	0.320758	0.321172
5	0.0213763	0.0158287	0.563107	0.436893	0.320435	0.321143

Configuration I generates a set of non-dominated solutions which are well distributed since it outperforms configuration D for the spacing metric. For the S metric, both configurations give very near results. The performance of D is slightly better than I for the contribution. From Table 1 and 2, we conclude that BEEDEA performs better on the spacing and the S metric than the configurations I and D. These results confirms that the two mechanisms are essential to better approximate the POF but don't show the efficiency of the diversification step to generate a set of diversified non-dominated solutions which are well uniformly distributed.

ZDT 6 problem:

This problem is a difficult multi-objective problem since the Pareto optimal front is non convex and non-uniformly distributed. Table 3 gives the results of BEEDEA and PSPEA for the ZDT6 problem on 5 processors. Table 3 shows that The BEEDEA outperforms the PSPEA for all the metrics. The BEEDEA is more efficient to approximate the POF. Table 4 gives the contribution of the BEEDEA and the PSPEA while varying the number of processors from 2 to 5. The contribution of the BEEDEA is larger than 62% for 2, 3 and 5 processors. This confirms the superiority of the proposed model to characterise the POF.

Table 3. Comparison of the BEEDEA and the PSPEA for the ZDT6 problem on 5 processors

	PSPEA	BEEDEA
Contribution	0,421821	0,578179
S Metric	0,5196546	0,4711904
Entropie	0,3359376	0,35660548
Spacing	0,01497213	0,01259775

Table 4. The contribution metric of the BEEDEA and the PSPEA for the ZDT6 problem while varying the number of processors from 2 to 5

number of processors	Contribution	
	PSPEA	BEEDEA
2	0.373397	0.626603
3	0.316156	0.683844
4	0.640351	0.359649
5	0.226531	0.773469

In Table 5 we study the influence of the number of processors on the performance of the BEEDEA. We remark that the contribution increases when the number of processors increases. The clustering algorithm partitions the global population into p (respectively 2p) clusters in the exploitation (respectively exploration) step. This number of clusters increases with the number of processors. This improves similarity intra-cluster and increases dissimilarity inter-clusters. Thus, exploitation and exploration are both well performed since neighbouring solutions are gathered in the same cluster.

Table 6 and 7 compare the performance of BEEDEA and configurations I and D for the ZDT6 problem on 5 processors. The BEEDEA outperforms the two configurations I and D when the algorithm runs without the exploration or the exploitation step. This confirms that the combination of the two steps performs better exploration of the search space.

Table 5. The contribution measure for the ZDT6 problem while varying the number of processors from 2 to 5 and setting the distribution interval to 15

number of processors	2	3	4	5
2		0.440909	0.759542	0.337128
3	0.559091		0.800971	0.389516
4	0.240458	0.199029		0.136674
5	0.662872	0.610484	0.863326	

Table 6. Comparison of the BEEDEA and the configuration I for the ZDT6 problem

	BEEDEA	I
Contribution	1	0
S Metric	0,4711904	4,895332
ENTROPIE	0,504699	0,17050098
SPACING	0,01259775	0,02835665

Table 7. Comparison of the BEEDEA and the configuration D for the ZDT6 problem

	BEEDEA	D
Contribution	1	0
S Metric	0,4711904	1,089886
ENTROPIE	0,504699	0,2006037
SPACING	0,01259775	0,04950688

5. Conclusion

MOEAs have been successfully used to solve real world MOPs. These algorithms require a huge amount of computational time and memory. To deal with this, several distributed schemes are proposed. In this paper, a new parallel elitist MOEA called balanced explore exploit clustering based parallel multi-objective evolutionary algorithm is proposed. It is based on a clustering algorithm. The proposed approach is applied to two numerical examples (ZDT2 and ZDT6). Its performance has been analysed in comparison to different versions in order to see the effect of the exploration and the exploitation mechanisms.

The combined use of exploitation and exploration is aimed to find high-quality solutions. From experimental results, we have seen that the proposed clustering based parallel MOEA have shown its efficiency on two benchmark problems. Additional experiments are necessary to distinguish between search space and objective space subdivisions (clustering).

REFERENCES

1. ALBA, TOMASSINI, M., **Parallelism and Evolutionary Algorithms**, IEEE Transactions on Evolutionary Computation, Vol. 6(5), 2002, pp. 443-462.
2. MAHMOOD, A., **A Hybrid Genetic Algorithm for Task Scheduling in Multiprocessor Real-Time Systems**, Studies in Informatics and Control, Vol. 9(3), June 2000.
3. BRANKE, J., H. SCHMECK, K. DEB, M. REDDY, **Parallelizing Multi-objective Evolutionary Algorithms: Cone Separation**, Congress on Evolutionary Computation, IEEE, 2004, pp. 1952-1957.
4. BRANKE, J., T. KAUFLEDER, H. SCHMECK, **Guidance in Evolutionary Multiobjective Optimization**, Advances in Engineering Software, vol. 32(6), 2001, pp. 499-508.
5. CANTÚ-PAZ, E., **A Survey of Parallel Genetic Algorithms**, technical report, Illinois Genetic Algorithms Lab., 1997.
6. DE TORO, F., J. ORTEGA, J. FERNANDEZ, A. DIAZ, **PSFGA: A**

- Parallel Genetic Algorithm for Multiobjective Optimization**, Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, IEEE, 2002, pp. 384-391.
7. DEB, K., **Multi-Objective Optimization Using Evolutionary Algorithms**, John Wiley & Sons, 2001.
 8. DEB, K., **Multiobjective Optimization using evolutionary algorithms**, Wiley, Chichester, UK, 2001.
 9. DEB, K., A. PRATAP, S. AGARWAL, T. MEYARIVAN, **A Fast and Elitist Multiobjective Genetic Algorithm: NSGA –II**, IEEE Transactions on Evolutionary Computation, vol. 6(3), 2002, pp. 182–197.
 10. DEB, K., P. ZOPE, A. JAIN, **Distributed Computing of Pareto-optimal Solutions with Evolutionary Algorithms**, In C.M. Fonseca, P.J. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, Evolutionary Multi-criterion Optimization, volume 2632 of LNCS, Springer, 2003, pp. 534-549.
 11. ERICKSON, M., A. MAYER, J. HORN, **The Niche Pareto Genetic Algorithm 2 Applied to the Design of Groundwater Remediation Systems**, In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, First International Conference on Evolutionary Multi-Criterion Optimization, Springer-Verlag, Lecture Notes in Computer Science No. 1993, 2000, pp. 681-695.
 12. ESSABRI, A., M. GZARA, T. LOUKIL, **Parallel Multi-Objective Evolutionary Algorithm with Multi-front Equitable Distribution**, The fifth International Conference on Grid and Cooperative Computing (GCC 2006). Changsha (China), IEEE, 21-23 October. 2006, pp. 241-244.
 13. FONSECA, C. M., P. J. FLEMING, **Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization**, Proceedings of the Fifth International Conference on Genetic Algorithms, ICGA-93, San Francisco, CA, Morgan Kaufmann, 1993, pp. 416-423.
 14. FOURMAN, M. P., **Compactation of Symbolic Layout using GAs**, In Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms and Their Applications. Lawrence Erlbaum, 1985, pp. 141-153.
 15. GOLDBERG, D. E., **Genetic Algorithms in Search, Optimization and Machine Learning**, Addison-Wesley, 1989.
 16. HAJELA, P., C. LIN, **Genetic Search Strategies in Multicriterion Optimal Design**, Structural Optimization, 4, 1992, pp. 99–107.
 17. HARBAOUI, D., R. KAMMARTI, M. KSOURI, P. BORNE, **A Genetic Algorithm for the Multi-Pickup and Delivery Problem with Time Windows**, Studies in Informatics and Control, vol. 18(2), 2009.
 18. HIROYASU, T., M. KANEKO, M. MIKI, **A Parallel Genetic Algorithm with Distributed Environment Scheme**, Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, vol. 2, 2000, pp. 619-625.
 19. HIROYASU, T., M. MIKI, S. WATANABE, **Distributed Genetic Algorithms with a New Sharing Approach in Multiobjective Optimization Problems**, 1999 Congress on Evolutionary Computation, Washington, D.C., July 1999, pp. 69–76.
 20. HORN, J., N. NAFPLIOTIS, D. E. GOLDBERG, **A Niche Pareto Genetic Algorithm for Multiobjective Optimization**, in Proceedings of the 1st IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Computation, Piscataway, vol. 1, 27–29 Jun 1994, pp. 82–87.
 21. <http://www.lifl.fr/OPAC/guimoo>.
 22. KAMIURA, J., T. HIROYASU, M. MIKI, S. WATANABE, T. SAKODA, **MOGADES: Multi-Objective Genetic Algorithm with Distributed**

- Environment Scheme**, Intelligent Systems Design and Applications (ISDA), 2002.
23. KAMMARTI, R., S. HAMMADI, P. BORNE, M. KSOURI, **A New Hybrid Evolutionary Approach for the Pickup and Delivery Problem with Time Windows**, SMC (2), 2004, pp. 1498-1503.
24. SCHAFFER, D., **Multiple Objective Optimisation with Vector Evaluated Genetic Algorithm**, Proceedings of the First International Conference on Genetic Algorithm, 1985, pp. 93-100.
25. SRINIVAS, N., K. DEB, **Multiobjective Optimization using Nondominated Sorting in Genetic Algorithms**, Evolutionary Computation, vol. 2(3), MIT Press, Cambridge, 1995, pp. 221-248.
26. STREICHERT D., H. ULMER, A. ZELL, **Parallelization of Multi-objective Evolutionary Algorithms using Clustering Algorithms**, Conference on Evolutionary Multi-Criterion Optimization (EMO 2005), LNCS series, V. 3410, SPRINGER, Guanajuato, Mexico, 9-11 March, 2005, pp. 92-107.
27. ZITZLER, E., E. DEB, L. THIELE, **Comparison of Multiobjective Evolutionary Algorithms: Empirical Results**, Evolutionary Computation Journal, vol. 8(2), 2000, pp. 125-148.
28. ZITZLER, E., M. LAUMANN, L. THIELE, **Spea2: Improving the Performance of the Strength Pareto Evolutionary Algorithm**, in technical report 103, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH) Zurich, 2001.
29. ZITZLER, E., THIELE, L., **Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach**, IEEE Transactions on Evolutionary Computation, Vol. 3(4), 1999, pp. 257-271.