On sharing information between sub-populations in MOEA/S

Ribeiro de Almeida, L.; Emmerich, M.T.M.; Da Silva Soares, A.; Woerle de Lima, T.; Bäck, T.M.T.; Preuss, M.; ... ; Trautmann, H.

On Sharing Information Between Sub-populations in MOEA/S

Lucas de Almeida Ribeiro[1,2(✉)], Michael Emmerich[3],
Anderson da Silva Soares[1], and Telma Woerle de Lima[1]

[1] Instituto de Informática, Universidade Federal de Goiás, Goiânia, Brazil
[2] Instituto Federal de Goiás, Goiânia, Brazil
lucas.ribeiro@ifg.edu.br
[3] Multiobjective Optimization and Decision Analysis Group, LIACS,
Leiden University, Niels Bohrweg 1, 2363CA Leiden, The Netherlands

Abstract. This work investigates the effect of information exchange in decomposition methods that work with multi-membered populations as sub-problems. As an algorithm framework, we use the *Multi-objective Evolutionary Algorithm based on Sub-populations* (MOEA/S). This algorithm uses parallel sub-populations that can exchange information via migration and/or recombination. For this work, each sub-population is constructed by a few weighted utility functions, grouped by distance between their weighting vectors. The question investigated in this paper is: How is the distance between sub-populations and the mechanism of information exchange influencing the performance of MOEA/S? The study considers two ways of transferring information: (1) migration of individuals, (2) recombination using parents from two different sub-populations. A matrix describing the linkage patterns between sub-populations governs migration and recombination mechanisms. This work conducts a systematic study using the multi-objective knapsack problem (MOKP) and multi-objective traveling salesperson (MOTSP) for two and three objectives test problems. The results motivated a restriction policy for sharing information. We compare an algorithm using this policy with other state-of-the-art MOEAs, including NSGA III, MOEA/D, and the previous version of MOEA/S.

Keywords: Decomposition-based multi-objective optimization · Cellular genetic algorithm · Sub-population based MOEAs · Migration operator · MOEA/S

1 Introduction

Multi-objective optimization is the task of finding solutions in a search space with the best quality concerning multiple objective functions. Decomposition-based multi-objective evolutionary optimization deals with these problems by defining a collective, population-based, search. The main idea of decomposition-based methods is to decompose the problem into sub-problems targeting

T. Bäck et al. (Eds.): PPSN 2020, LNCS 12270, pp. 171–185, 2020.
https://doi.org/10.1007/978-3-030-58115-2_12

different regions on the Pareto front. The search is done simultaneously and while continuously exchanging information between the sub-populations [7,10,16].

One crucial matter in decomposition-based methods for multi-objective problems is how to exchange information among sub-populations to speed up the convergence to the Pareto front based on shared search and to improve Pareto front coverage. The mechanism used for this purpose covers mating individuals from different sub-populations and migrating individuals among sub-populations.

Zhang et al. propose a straightforward and commonly used implementation of decomposition-based methods. [29], called the Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D). MOEA/D explicitly decomposes the multi-objective optimization problem into N (size of the population) scalar optimization sub-problems. Then, MOEA/D optimizes the N sub-problems simultaneously. Each sub-problem is optimized by only using information from its neighboring sub-problems, where the neighborhood is defined a priori based on the neighborhood of reference directions. MOEA/D uses the same aggregation function for all sub-problems; a unique combination of weighting vectors defines each sub-problem. Thus, the neighborhood of a sub-problem is assumed as the neighborhood of its weighting vector.

Another family of decomposition-based methods, used in the literature, splits the population into several sub-populations where each one of them can use a distinct multi-objective strategy [2–5,9,11,22–25]. In this work, we formulate this approach as MOEA/S (Multi-objective Evolutionary Algorithm based on Sub-populations). Although this method has obtained good results, only a very general rule is defined for exchanging information among sub-populations. In contrast to MOEA/D, which uses a neighborhood definition, the simple rule is that every sub-population is exchanging information with every other sub-population at the same rate.

A commonly applied decomposition-based algorithm is the new version of the Non-dominated Sorting Genetic Algorithm (NSGA-III) [7], which places reference points on a simplex the size of which is adapted according to the current best information on the boundaries of the true Pareto front. NSGA-III does not take into account neighborhood among sub-populations in exchanging information, although it uses a niching method for selection.

Murata et al. [17] studies the behavior of restricting mating and replacement based on the neighborhood in a cellular version of Multi-Objective Genetic Algorithm (MOGA). Using different neighborhood sizes, they conclude that neither the closest neighbor nor the farthest sub-population is the best option in sharing information (using mating and replacement). Whether such strategies are better than not sharing information remains open in their analysis.

Ishibuchi et al. [12] studies the use of different size of neighborhoods (Ts) for mating and replacement selection in many-objective problems in MOEA/D. They obtain as a result that an appropriate specification of the two neighborhoods is problem-dependent. But in all the cases, a small neighborhood for replacement might lead to a well-distributed Pareto front, in many-objective problems. Wang et al. [28] suggest new replacement strategies where the solution is compared in all weight vectors and replace the solutions in the neighborhood of its best suitable weight vector.

These previous works have shown good results in exploring the relation between neighborhood size and performance. Thus, the exploitation of information obtained in similar sub-problems is useful to improve the speed of convergence to the Pareto front. Nevertheless, these results are not extended for non-cellular approaches, since the neighborhood of one sub-population with multiple individuals is not as easy to define as the neighborhood of a singleton sub-population as it is used in MOEA/D and c-MOGA.

Our work will investigate the effectiveness of exchanging information between sub-populations based on their distance using the non-cellular decomposition method, MOEA/S. MOEA/S decomposes a problem into N scalar optimization sub-problems. Each sub-problem is solved simultaneously using a population-based multi-objective evolutionary algorithms (MOEA) - in accordance with previous research, the N populations used by these MOEAs will be called sub-populations. In contrast to MOEA/D, each sub-population can consist, in general, of more than one individual. In order to exchange information, a connection between sub-populations must be established. The connection definition is based on the distance between the centroids of the sub-populations. Different operators for information exchange will be compared for MOEA/S in this paper.

This paper is organized as follows: Sect. 2 introduces MOEA/S and explains the main conceptual ideas of the method. Section 3 shows experimental setup; Sect. 4 explores the results on test problems and in Sect. 5 the paper is concluded with a summary of our main findings[1].

2 Methods

2.1 MOEA/S Algorithm

The Multi-objective Evolutionary Algorithm based on Sub-populations (MOEA/S) is a decomposition-based MOEA which supports non-singleton sub-population based MOEAs to solve, simultaneously, the sub-problems of a problem decomposition. In principle, each sub-problem can be solved by a different MOEA, in terms of the selection processes. A global 'master algorithm' controls the interplay and information exchange between the MOEAs that address sub-problems.

MOEA/S splits the (global) population into a constant number of μ sub-populations, which are managed by different selection processes. One can design a process using Pareto based strategies, indicator-based methods, scalarization based algorithms, and so on. To be eligible as a MOEA, for solving a sub-problem in MOEA/S, the selection operator must obey a particular framework. The framework interface requires: limited population size of at most N^i_{limit} or $N^i (i = 1 \dots \mu)$ individuals; moreover, it must define a method for mating selection; and a method for environmental (or truncation) selection.

In MOEA/S, it is an essential principle that sub-problems are not solved independently, but in general, it is possible to exchange information between

[1] Additional data is made available in the web-repository http://moda.liacs.nl.

sub-populations. The idea is, roughly speaking, to exploit synergies between different sub-problem solution processes.

Sub-populations can exchange information in two stages: the first stage, the mating stage, is using the mating operator. Via a mating matrix, a coupling between the sub-populations is established. The rows (index $i \in \{1, \dots, \mu\}$) indicate the populations in the mating pool (deme) of the i-th sub-population P_i. Secondly, in the migration stage, a migration matrix (destination matrix) is set up to decide to which other populations, individuals of sub-population P_i can migrate (for each $i = 1, ..., \mu$).

In summary, MOEA/S contains a list of sub-populations, $(P_1, \dots, P_\mu)$, each of which containing a limited number $N_i, i = 1, ..., \mu$ of individuals, a method for selecting parents (mating selection), and a method for discarding or selecting individuals; a population (P $= \bigcup_{i=1}^{\mu} P_i$); a structure which stores the connections between sub-populations for mating (deme - $M \in \mathbb{B}^{\mu \times \mu}$); and for destination sub-populations (destination matrix - $D \in \mathbb{B}^{\mu \times \mu}$) used in the environmental selection; environmental selection maintains an adjacency matrix to associate each individual with its sub-populations (adjacency matrix - $A \in \mathbb{B}^{\xi \times \mu}$, where $\xi = |P|$); a method for creating new individuals; and a method for initialization.

The MOEA/S procedure starts with the initialization of the sub-population structure. This phase distributes all individuals from the initial population into the sub-populations. The evolutionary loop consists of: (1) *selection of the parents* (or mating pool), (2) *creating new individuals by mutation and crossover operators*, (3) *environmental selection*, (4) *migration.* In more detail:

(1) select $p1$ as the first sub-population. Then, the first parent (s^{p1}) is selected from a designated sub-population (P_{p1}) and the second one (s^{p2}) is selected from a population P_{p2}. Index $p2$ is chosen according to the mating pool of P_{p1} defined by the mating matrix.
(2) generate new individuals (s^{new1} and s^{new2}) by crossover and mutation operators from s^{p1} and s^{p2}
(3) evaluates the new individuals in the sub-populations of their parents.
(4) migrate the new individuals to the destination sub-populations of their parents' sub-populations, according to the migration matrix.

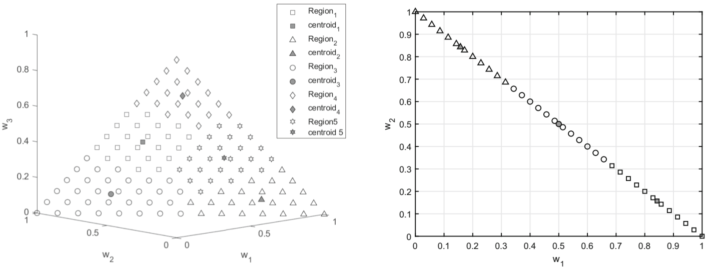
An individual s^{new} is accepted in the sub-population P_i if the size of P_i satisfies $|P_i| < N_i$ or in case $|P_i| = N_i$ it can be chosen by the selection of the destination sub-population, for instance, a tournament selection. In the latter case, an individual of the destination sub-population s^{new} replaces s^{old} in P_i.

2.2 MOEA/S Instance

Next, we will discuss the specific instance of MOEA/S used in this paper, which targets different regions on the Pareto front by different sub-populations:

First, N (size of the population) scalar optimization functions are defined differed by their weighting vector, as in MOEA/D. Each one of these functions is associated with one sub-problem (R^i, $1, ..., N$). Then, sub-problems are clustered into sub-populations, further called Region sub-populations (R^i, $1, ..., \mu$), using their weighting vectors. Thus, each individual in the population is associated with one sub-problem, and its fitness is assessed by one utility function.

Figure 1 exemplifies how weighting vectors distributions for two and three-objective spaces are spatially located. Figure 1(a) presents the regions on 3-objective spaces with 120 points and 5 Regions. Figure 1(b) presents the regions on 2-objective spaces, with 20 points and 3 regions. The clustering method is k -means clustering [1] which groups each sample around a centroid, where here the number of clusters corresponds to the number of regions.



(a) Regions on 3-objectives

(b) Regions on 2-objectives

Fig. 1. Distribution of solution targets based on their weighting vectors.

As weighting or utility function Chebychev scalarization is used: Given a solution s, an lower bound point s^{lb} (where $s_i^{lb} = min(f_i(X'))$, being X' the explored search space so far)), and a weight vector $(w_1^1, \ldots, w_k^1)$ associated to a problem with k objective functions $(f_1, \ldots, f_k)$, then the Chebychev utility function reads $u_{Cheb} = \max_{i=1}^{k} w_i(f_i(s) - s_i^{lb})$.

The mating selection in a Region Sub-population selects one individual based on binary tournament selection. This selection method picks two individuals randomly and chooses the one with the best fitness, where the fitness of a solution is its fitness-value regarding the sub-problem which it is associated.

The environmental selection in a Region Sub-population (R^i) for a new solution (s^{new}) is accomplished by evaluating s^{new} in each one of the sub-problems from this region. This process goes until all sub-problems ($r^j \in R^i$) have been visited, or s^{new} is accepted by some sub-problem in R^i.

For each Region Sub-population (say i) a list of all other regions (say $j, j \neq i$) is created and the list is sorted by the distance to the centroid of R^i. This way we establish nearest neighboring region, second neighboring regions of different radius. This collection of lists will be treated as a matrix $L \in \mathbb{B}^{\mu \times \mu}$ where the $L(i, j)$ equals to the ranking distance between i and J. This structure is used to construct matrices D and M.

2.3 Sharing Information by Migration

Locality is a fundamental resource when using guided search methods, assuming that the structure of the fitness landscape leads search algorithms to high quality solutions [19]. Where low locality degenerate the performance of the search algorithm in a random search [21]. This principle provides two results: small changes in solutions cause small changes in fitness values; solutions with high fitness values are spatially localized.

The creation of a new offspring consists of the subsequent application of the recombination operator and the mutation operator:

- Recombination operators perform search exchanging information among solutions. In this operator, the information content of multiple individuals (normally two) are combined in order to generate a new individual with mutual information from its parents. Recombination operators generate offspring, where the distances between offspring and parents are usually equal to or smaller than the distance between parents.
- A mutation operator generates a solution s^{new} from s^{old} by a small random change in s^{old}. Mutation operators do not use the neighborhood lists.

Together, a search step combining recombination and subsequent mutation produces an offspring in a neighbourhood, which encloses the parents. A search step is useful if it generates a solution in an area of interest regarding a subpopulation. Once two sub-populations can overlap a common area of interest, sharing offspring solutions can be beneficial. The process of sharing offspring between sub-populations is known as migration.

This migration operator demands a topology defining links between source and target sub-populations. Sprave [26] presents a formal model of population structures in evolutionary algorithms based on hypergraphs. This model allows using an individual hypergraph matrix as migration topology or mating selection topology. Here, we use this idea defining the migration topology by means of destination matrix (D) where $d_{ij} = 1$ if i is a source sub-population and j is a target sub-population, and $d_{ij} = 0$ otherwise. In this work only newly created individuals are submitted for migration.

2.4 Sharing Information by Recombination

Next to migration, MOEA/S also allows sharing information by mating parents from different sub-populations. An essential step in recombination methods is selecting the right combination of mates in order to generate useful descendants. This step is called mating selection. The problem in mating selection can be stated as: given a first parent solution s^{p1} from P_{p1}, which other sub-population P_{p2} should be selected in order to find a good matching (s^{p2}) for the first parent?

Multi-objective problems deal with highly conflicting objectives, and hence the search in each sub-population leads to different specialized region. Therefore, combining solutions from sub-populations located on extremely different parts of the Pareto front becomes unreasonable as they evolve. In contrast, solutions in similar sub-problems have similar information content; combining these solutions leads to exploiting small regions, degrading the search in the first generations.

3 Experiments

The multi-objective 0-1 knapsack problem based on [31] is defined for 2 and 3 objectives with 500 items, we call it MOKP-2 or MOKP-3 according to the number of objectives. The second problem is a multi-objective formulation of traveling salesperson problem (MOTSP) [6] defined for 2 and 3 objectives with 30 cities, we call it MOTSP-2 or MOTSP-3 according to the number of objectives. These problems were chosen because in these problems the similarity of solutions reflects to some extend the similarity of the obtained results. As opposed to many problems in continuous multi-objective optimization, such as ZDT [30] and DTLZ [8], spread and convergence are both influenced by all variables. In other words, there is no separation of variables that influence only spread or only convergence. Moreover, the problems have practical relevance and are structurally similar to real world problems.

The hypervolume indicator and R2-indicator were used to assess the performance of the population. The hypervolume indicator has been the most used quality indicator in the performance assessment of Pareto front approximations [18,20]. The hypervolume indicator measures the size of the region dominated by an approximation set [31], and bound from above by a reference point. R2-indicator is defined as an integral over a weight space for a family of distance to a reference point utility function (typically weighted Chebychev distance to the ideal point). Thus, R2-indicator is very suitable for decomposition problems (which uses also utility functions).

In this paper we also propose two metrics: number of useful migrations (α) and number of useful mating (β). α results from: given distance rank[2], denoted with ($\rho \in 1, \ldots, \mu$), α_ρ counts how many useful migrations occurs between sub-populations in distance ρ; thus, for instance, α_1 counts how many individuals from R^i are accepted by R^j, with R^j being the nearest neighboring sub-population from R^i. Second metric, β states: given a distance rank ρ ($\in 1, \ldots, \mu$), β_ρ counts useful offspring resulted from mating between sub-populations in this distance; thus, β_1 counts how many individuals are accepted (in any sub-population) from matches between R^i and R^j, once R^j is the nearest neighboring sub-population from R^i; β_0 counts how many individuals are accepted (in any sub-population) from a mating of parents from the same sub-population.

3.1 Experiment Settings

The MOEA/S setting in the research study on the benefits of sharing information (Experiments 1 and 2) is given by: μ - *number of sub-populations* equals

[2] Note, as a detail, that in the case of ties, that is two sub-populations share the same distance, the distance rank will be randomly assigned.

to 6; N - *number of individuals* set as 36; *set of problems* defined as MOKP-2, MOKP-3, MOTSP-2, MOTSP-3; and *number of generations* is (120000/N)-2.

For MOKP problems, we used binary representation, one-point crossover as in [31], and 2/500 bit-flip mutation rate. For MOTSP problems, we used permutation representation with order crossover and swap mutation. Crossover and mutation rates of 1. The presented results are average performance metrics obtained by the populations at a given time; this average considers 20 runs of the algorithm (generation vs. quality measure). Thus, we can study the sharing process between the sub-populations according to the time (generation number).

Experiment 1: Sharing Information by Migration Between Independent Sub-populations. First experiment explores the relationship between neighborhood of a sub-population and the effectiveness of sharing its descendants by **migration**. Destination Matrix: $D(i, j) = 1$, for all i and j. The mating matrix now reads $M(i, j) = 1$ if $i = j$; 0, otherwise. We compare the different α over the generations. That is, we assess the success that is attributed to migration of different radius. For statistical smoothing purposes, we report cumulative values of α over ranges of distance ranks.

Experiment 2: Sharing Information by Recombination. In the second experiment there is no restriction on mating or migration selection process. The destination matrix is set to $D(i, j) = 1$, for all i and j. Thus, the mating matrix reads $M(i, j) = 1$, for all i and j. The other parameters are set as in Sect. 3.1. This experiment aims at understanding how the distance between sub-populations of parents is related with producing successful offspring. We used a scheme selection that guarantees all β range are assessed. For statistical smoothing purposes, we report cumulative values of β over ranges of distance ranks.

Experiment 3: Using Local vs. Global Sharing in MOEA/S. Last experiment compares three MOEA/S designs ($MOEAS_{can}$, $MOEAS_0$ and $MOEAS_1$) with MOEA/D [29] and NSGA-III [7] implementations found in PlatEMO [27]. Here we compare approaches with global and local sharing policies. All algorithms in this experiment use the same maximum size of population and search operators. Here: $N = 120$ is the (maximum) population size. As specific parameters MOEA/D uses neighborhood size $T = N/10$; NSGA-III uses N accumulation points; and MOEA/S implementations work with $\mu = 10$ sub-populations.

The MOEA/S implementations are detailed as follows:

- **Global Sharing** $MOEAS_{can}$ ('can' stands for canonical) defines no restriction over mating parents from different sub-populations. A new solution can migrate to all sub-populations. M and D are set as Experiment 2.
- **No Sharing** $MOEAS_0$ each sub-population works independently and there is no sharing, i.e. the sub-populations work in parallel without migration; M is as in Experiment 1 (no mating across sub-populations) and $D = M$.

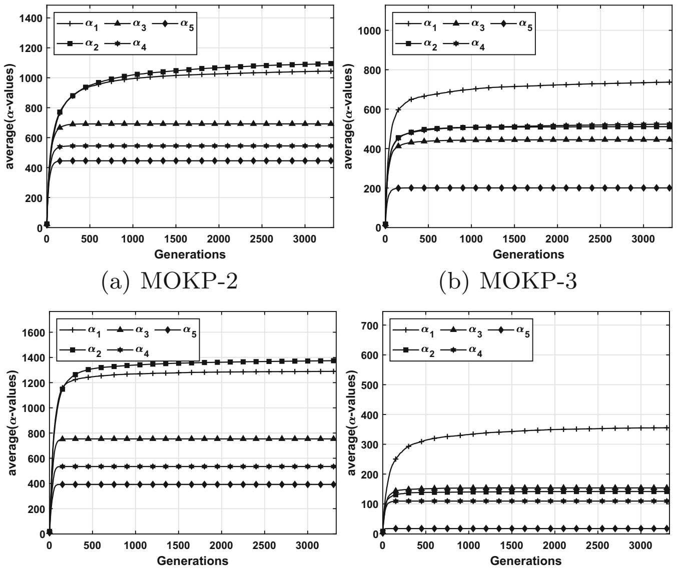
- **Local Sharing** In $MOEAS_1$ the distance between two sources of parents is restricted to 1 and solutions can migrate only for the three closest neighborhoods; thus, $M(i,j) = 1$ if $L(i,j) \leq 1$, 0 otherwise; and $D(i,j) = 1$ if $L(i,j) \leq 3$, 0 otherwise. Here L is the sorting matrix defined in Sect. 2.2.

The reference points for Hypervolume-indicator was set as (26098, 28367) and (27576, 27483, 27367) for MOKP-2 and MOKP-3 test problem, respectively; for MOTSP-2 and MOTSP-3 problems it was set as (296.88, 295.32) and (288.8, 288.54, 284.15), respectively. R2 was implemented using Chebychev scalarization based utility function with the N (maximun size of the population) number of points. As Hypervolume-based and R2 indicators have obtained similar results (same ranking position when comparing the algorithms), therefore, we just show Hypervolume-indicator.

4 Results and Discussion

4.1 Sharing Information by Migration Between Sub-populations

Figure 2 illustrates the success rate of sharing information, by migration operator, between sub-populations based on their distance.



(a) MOKP-2

(b) MOKP-3

(c) MOTSP-2

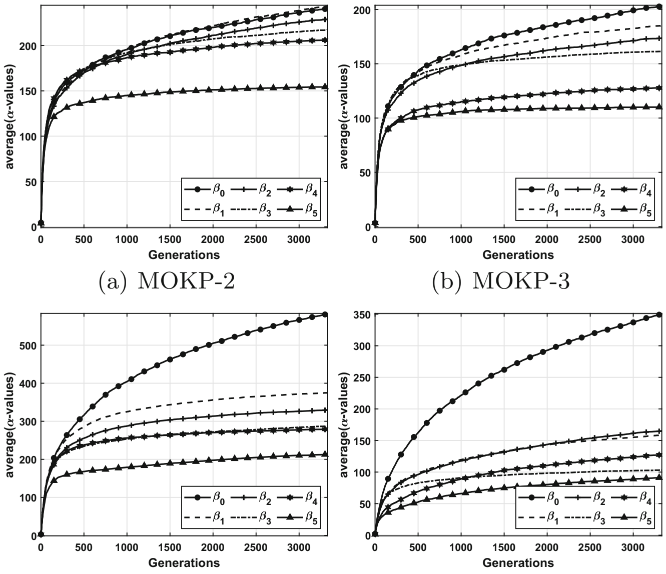
(d) MOTSP-3

Fig. 2. Migration rate success grouped by distance between source and destination. Average α is the cumulative counting of accepted offsprings in a given distance.

Migrating information with the first neighbour is the best option for 3-objective problems, and it is the second best option for 2-objective problems (the best option is in the 40% closest neighboring sub-populations). In all cases, sharing information with the furthest sub-population is unlikely to be successful, in particular in later stages of the search when it becomes specialized. Although the success rate is low, migration does not affect the generation, i.e., a bad migration try is not a waste in execution count. Thus, without taking care the effort of validating a solution, sharing with all sub-populations is the best option.

4.2 Sharing Information by Recombination Between Sub-populations

The second study (Fig. 3) analyzes the behavior of the population quality during the evolutionary process when applying both of the operators, recombination and migration. This results reinforce the idea from Ishibuchi and Shibata [13–15] about using similarity indicators in mating selection. Crossing individuals between sub-populations can be as useful as crossing neighboring individuals. However, as the search progresses, the probability of generating good offspring by crossbreeding sub-populations decreases. Crossbreeding with the nearest neighbouring sub-populations remains successful also in the later stage of search.



(a) MOKP-2

(b) MOKP-3

(c) MOTSP-2

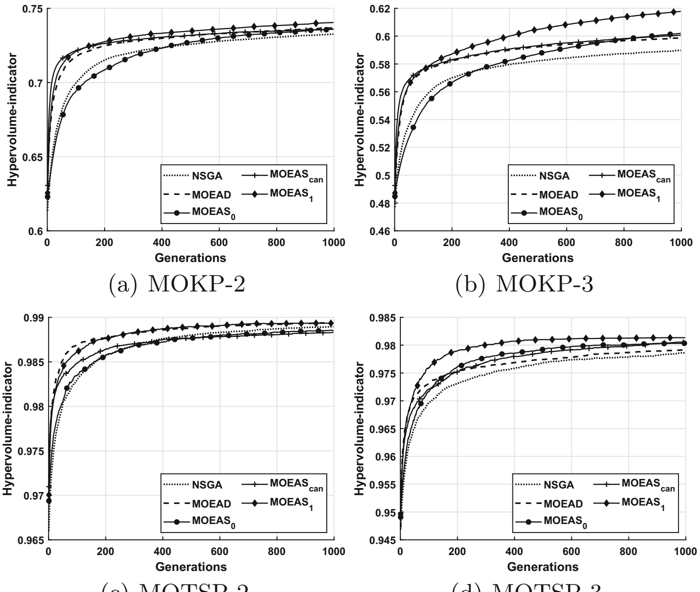
(d) MOTSP-3

Fig. 3. Mating rate effectiveness grouped by distance between source sub-population of parent 1 and source of parent 2. Average β is a cumulative value during the search.

As result from Fig. 3, the highest probability of generating useful offspring is obtained by crossing individuals from the same region. This result is shown in both problems. The only exception is Fig. 3(a) where the recombination with the first neighboring region has similar (to better) behavior. As the search progresses the probability of finding useful individuals from apart sub-populations decreases. The best mating selection (between sub-populations) scenario occurs in the first neighborhood. Only β_1 and β_2 continue increasing over time in all test cases.

4.3 Using Local vs. Global Sharing in MOEA/S

From previous results (Sects. 4.1 and 4.2), most of information needed for improving search performance in a sub-population comes from the nearest sub-populations. Thus, the last experiment studies the behavior of the evolutionary process when defining mating and migration rules, by comparing global sharing ($MOEAS_{can}$), no sharing or independent sub-populations ($MOEAS_0$), and local sharing ($MOEAS_1$) versions of MOEA/S. We also compare its behavior with MOEA/D and NSGA III implementations. Figure 4 presents the search behavior of these algorithms regarding Hypervolume indicator.



(a) MOKP-2

(b) MOKP-3

(c) MOTSP-2

(d) MOTSP-3

Fig. 4. Performance of MOEA/S using mating and migration restriction compared with other MOEAs implementations including NSGA III and MOEA/D.

As result, from Fig. 4, sharing information has been demonstrated as the right choice for improving the convergence rate on MOEA/S. Sharing information with all sub-populations is one of the best options in the beginning of the search, where $MOEAS_{can}$ is the best option for MOKP test problems (Fig. 4(a) and 4(b)) until ca. generation 100. However, as the search progresses $MOEAS_0$ and $MOEAS_1$ continue improving search performance, while MOEA/D and $MOEAS_{can}$ prematurely converge. Only in Fig. 4(c) MOEA/D has a similar performance when compared to $MOEAS_1$ even after generation 500 (no significant difference by Wilcoxon rank sum test, $p = 0.05$).

Once MOTSP-3 takes advantage on neighboring recombination (as shown in experiments Sects. 4.1 and 4.2), Fig. 4(d), $MOEAS_0$ shows its best ranking performance. NSGA-III performs better with two-objective problems as compared to three-objective test problems. *Thus, sharing information can be considered beneficial for mating and migration selection.*

5 Conclusion and Outlook

Our study has investigated sharing in multi-objective optimization across sub-populations that explore different regions of the Pareto front. Both, sharing by migration and by mating has found to be useful tool for improvement of combinatorial multi-objective optimization. Diversity is achieved by exchanging information between dissimilar sub-populations, which influences the performance of the firsts generations. On the other hand, focusing on similar sub-populations can improve exploitation in the search. Consequently, mating neighboring parents leads to better final results. Another important finding is, that the radius of sharing and the type of sharing has a crucial influence on its beneficial effect. Moreover, long radii have found to more benefit early stages of search, whereas in later stages short, but non-zero, radii for sharing are more beneficial.

The study points out and confirms some interesting phenomena regarding sharing and paves the way to future work taking these novel findings into account: Adaptive selection schemes could be considered once the effectiveness of distance-based migration, and mating selection depends on the stage of the search. In the final stages, there is no need to migrate solutions or mating solutions between sub-populations. The selection scheme proposed by Ishibuchi [15] can be the right choice if we consider the panmictic population. However, this selection scheme is not extensible for parallel populations.

Since our study suggests that mating selection is highly related with neighborhood of solutions, future work on designing MOEA/S should take neighborhood adaptation measures for mating into account. Moreover, there is room for discussion on neighborhood/deme representations using hypergraphs (see Sprave [26]). In particular, such considerations might be of relevance for theoretical analysis using Markov chain techniques.

Acknowledgments. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. We thank LIACS for hosting and promoting the collaboration, which resulted in this paper. Particularly, we thank NACO and MODA groups at LIACS and Laboratory of Modern Heuristics at INF/UFG by the discussions and background.

References

1. Arthur, D., Vassilvitskii, S.: K-means++: the advantages of careful seeding. In: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (2007)
2. Brasil, C.R.S., Delbem, A.C.B., da Silva, F.L.B.: Multiobjective evolutionary algorithm with many tables for purely ab initio protein structure prediction. J. Comput. Chem. **34**(20), 1719–1734 (2013)
3. Camillo, M.H.M., et al.: Combining exhaustive search and multi-objective evolutionary algorithm for service restoration in large-scale distribution systems. Electric Power Syst. Res. **134**, 1–8 (2016)
4. Camillo, M.H.M., et al.: Validation of a methodology for service restoration on a real Brazilian distribution system. In: 2014 IEEE PES Transmission & Distribution Conference and Exposition-Latin America (PES T&D-LA), pp. 1–6. IEEE (2014)
5. Camillo, M.H.M., et al.: A multi-objective evolutionary algorithm with efficient data structure and heuristic initialization for fault service restoration. Procedia Comput. Sci. **80**, 2367–2371 (2016)
6. Corne, D.W., Knowles, J.D.: Techniques for highly multiobjective optimisation: some nondominated points are better than others. In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, GECCO 2007, pp. 773–780. Association for Computing Machinery, New York (2007)
7. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. IEEE Trans. Evol. Comput. **18**(4), 577–601 (2014)
8. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: Abraham, A., Jain, L., Goldberg, R. (eds.) Evolutionary Multiobjective Optimization, pp. 105–145. Springer, London (2005). https://doi.org/10.1007/1-84628-137-7_6
9. Delbem, A.C.B., de Carvalho, A.C.P.D.L.F., Bretas, N.G.: Main chain representation for evolutionary algorithms applied to distribution system reconfiguration. IEEE Trans. Power Syst. **20**(1), 425–436 (2005)
10. Fonseca, C.M., Fleming, P.J.: Genetic algorithms for multiobjective optimization: formulation: discussion and generalization. In: Proceedings of the 5th International Conference on Genetic Algorithms, San Francisco, CA, USA, pp. 416–423. Morgan Kaufmann Publishers Inc. (1993)
11. Gois, M.M., Sanches, D.S., Martins, J., Junior, J.B.A.L., Delbem, A.C.B.: Multi-objective evolutionary algorithm with node-depth encoding and strength pareto for service restoration in large-scale distribution systems. In: Purshouse, R.C., Fleming, P.J., Fonseca, C.M., Greco, S., Shaw, J. (eds.) EMO 2013. LNCS, vol. 7811, pp. 771–786. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37140-0_57
12. Ishibuchi, H., Akedo, N., Nojima, Y.: Relation between neighborhood size and MOEA/D performance on many-objective problems. In: Purshouse, R.C., Fleming, P.J., Fonseca, C.M., Greco, S., Shaw, J. (eds.) EMO 2013. LNCS, vol. 7811, pp. 459–474. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37140-0_35
13. Ishibuchi, H., Shibata, Y.: An empirical study on the effect of mating restriction on the search ability of EMO algorithms. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Thiele, L., Deb, K. (eds.) EMO 2003. LNCS, vol. 2632, pp. 433–447. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36970-8_31

14. Ishibuchi, H., Shibata, Y.: A similarity-based mating scheme for evolutionary multiobjective optimization. In: Cantú-Paz, E., et al. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 1065–1076. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-45105-6_116
15. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Choosing extreme parents for diversity improvement in evolutionary multiobjective optimization algorithms. In: 2007 IEEE International Conference on Systems, Man and Cybernetics, ISIC, pp. 1946–1951. IEEE (2007)
16. Jain, H., Deb, K.: An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: handling constraints and extending to an adaptive approach. IEEE Trans. Evol. Comput. **18**(4), 602–622 (2014)
17. Murata, T., Ishibuchi, H., Gen, M.: Specification of genetic search directions in cellular multi-objective genetic algorithms. In: Zitzler, E., Thiele, L., Deb, K., Coello Coello, C.A., Corne, D. (eds.) EMO 2001. LNCS, vol. 1993, pp. 82–95. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44719-9_6
18. Okabe, T., Jin, Y., Sendhoff, B.: A critical survey of performance indices for multiobjective optimisation. In: The 2003 Congress on Evolutionary Computation, CEC 2003, vol. 2, pp. 878–885. IEEE (2003)
19. Raidl, G.R., Gottlieb, J.R.: Empirical analysis of locality, heritability and heuristic bias in evolutionary algorithms: a case study for the multidimensional knapsack problem. Evol. Comput. **13**(4), 441–475 (2005)
20. Riquelme, N., Von Lücken, C., Baran, B.: Performance metrics in multi-objective optimization. In: 2015 Latin American Computing Conference (CLEI), pp. 1–11. IEEE (2015)
21. Rothlauf, F.: Design of Modern Heuristics: Principles and Application. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-540-72962-4
22. Sanches, D.S., Mazucato, S.C., Castoldi, M.F., Delbem, A., London Jr., J.B.: Combining subpopulation tables, non-dominated solutions and strength pareto of MOEAs to treat service restoration problem in large-scale distribution systems. In: IECON 2013–39th Annual Conference of the IEEE Industrial Electronics Society, pp. 1986–1991, November 2013
23. Sanches, D.S., et al.: Multiobjective evolutionary algorithm with a discrete differential mutation operator developed for service restoration in distribution systems. Int. J. Electr. Power Energy Syst. **62**, 700–711 (2014)
24. Sanches, D.S., Mansour, M., London Jr., J.B., Delbem, A., Santos, A.C.: Integrating relevant aspects of MOEAs to solve loss reduction problem in large-scale distribution systems. In: 2011 IEEE PES Trondheim PowerTech: The Power of Technology for a Sustainable Society, POWERTECH 2011, June 2011
25. Santos, A., Delbem, A., London, J.B., Bretas, N.: Node-depth encoding and multiobjective evolutionary algorithm applied to large-scale distribution system reconfiguration. IEEE Trans. Power Syst. **25**(3), 1254–1265 (2010)
26. Sprave, J.: A unified model of non-panmictic population structures in evolutionary algorithms. In: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), vol. 2, pp. 1384–1391 (1999)
27. Tian, Y., Cheng, R., Zhang, X., Jin, Y.: Platemo: a matlab platform for evolutionary multi-objective optimization [educational forum]. IEEE Comput. Intell. Mag. **12**(4), 73–87 (2017)
28. Wang, Z., Zhang, Q., Zhou, A., Gong, M., Jiao, L.: Adaptive replacement strategies for MOEA/D. IEEE Trans. Cybern. **46**(2), 474–486 (2016)

29. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007)
30. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: empirical results. Evol. Comput. **8**(2), 173–195 (2000)
31. 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)