MOEAs Are Stuck in a Different Area at a Time

Li, Miqing; Han, Xiaofeng; Chu, Xiaochen

DOI:
10.1145/3583131.3590447

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (Harvard):
Li, M, Han, X & Chu, X 2023, MOEAs Are Stuck in a Different Area at a Time. in *GECCO '23: Proceedings of the Genetic and Evolutionary Computation Conference.* GECCO: Genetic and Evolutionary Computation Conference, Association for Computing Machinery (ACM), pp. 303-311, GECCO '23: Genetic and Evolutionary Computation Conference, Lisbon, Portugal, 15/07/23. https://doi.org/10.1145/3583131.3590447

MOEAs Are Stuck in a Different Area at a Time

Miqing Li*
m.li.8@bham.ac.uk
University of Birmingham
Birmingham, United Kingdom

Xiaofeng Han
hanx63520@gmail.com
Harbin Institute of Technology
Harbin, China

Xiaochen Chu
chuxiaochen95@163.com
Harbin Institute of Technology
Harbin, China

ABSTRACT

In this paper, we show that when dealing with multi-objective combinatorial optimisation problems, the search, in different executions of a multi-objective evolutionary algorithm (MOEA), e.g., NSGA-II, tends to stagnate in different areas in the search space. In other words, the final populations obtained by an MOEA under multiple executions, which can be very close in the objective space, are located far away from each other in the search space. This phenomenon becomes more apparent with the increase of some type of problem complexity (e.g., the ruggedness level of problem landscape). Interestingly, the phenomenon only happens to combinatorial optimisation problems, but not to continuous ones. In this study, we consider three well-established MOEAs (NSGA-II, SMS-EMOA and MOEA/D) on two representative combinatorial optimisation problems (NK-landscape and TSP) and on two commonly used continuous problem suites (DTLZ and WFG). Experimental results show a clear difference between multi-objective combinatorial and continuous problems and suggest a need of more efforts to be put on developing effective MOEAs for combinatorial problems.

CCS CONCEPTS

• **Mathematics of computing** → **Combinatorics**; **Combinatoric problems**; • **Applied computing** → **Multi-criterion optimization and decision-making**.

KEYWORDS

Combinatorial optimisation, multi-objective optimisation, evolutionary algorithms, distance metric, search space

ACM Reference Format:
Miqing Li, Xiaofeng Han, and Xiaochen Chu. 2023. MOEAs Are Stuck in a Different Area at a Time. In *Proceedings of The Genetic and Evolutionary Computation Conference 2023 (GECCO '23).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3583131.3590447

1 INTRODUCTION

In real life, many optimisation scenarios are of multiple objectives which need to be considered simultaneously. Based on the type of the variables, multi-objective optimisation problems (MOPs) can be divided into combinatorial MOPs and continuous MOPs. In the former, some or all of the problem's variables belong to a discrete set, while in the latter, the variables are allowed to take on any real number within a range of values. A prominent feature of MOPs is that there is no single optimal solution which can achieve the best on all the objectives, but rather a set of trade-off solutions which are not comparable to each other, termed as Pareto optimal solutions.

Typically, in an MOP the number of Pareto optimal solutions can be prohibitively large (exponentially for combinatorial MOPs and infinitely for continuous ones). It is not practical to find all Pareto optimal solutions of an MOP. Therefore, the goal of a multi-objective optimiser is to find a good approximation of the Pareto optimal set, so that the decision maker can choose their preferred solution to deploy. With this goal, evolutionary algorithms, a class of global search algorithms inspired by biological evolution [4], have been considered to be a good choice for dealing with MOPs. Their population-based search nature can approximate the Pareto optimal set within one execution, with each individual in the population representing a different trade-off between the objectives.

Over the last two decades, there are numerous evolutionary algorithms developed for MOPs, termed as multi-objective evolutionary algorithms (MOEAs). In general, existing MOEAs can be divided into three classes based on their way of comparing and ranking solutions: Pareto-based MOEAs, indicator-based MOEAs and decomposition-based MOEAs [17]. Despite the variety, they all serve the same purpose of finding a set of diversified solutions that can well approximate the problem's Pareto optimal set.

In this paper, we, however, present that MOEAs (whatever Pareto -based, indicator-based or decomposition-based) can be trapped during the search, and more importantly, we present that MOEAs in different executions stagnate in very different areas in the search space. In other words, the obtained populations, which can be similar in terms of their images in the objective space, are located far away from each other in the search space. Interestingly, this phenomenon only occurs to combinatorial MOPs but not to continuous ones. This highlights the difference between the two types of MOPs and suggests a need of more efforts put on developing effective MOEAs for combinatorial MOPs.

Much effort on combinatorial multi-objective optimisation is to develop effective local/hybrid search algorithms [7, 16]. This includes integrating local search into MOEAs as inner components (e.g., in [23, 25]) or used as post-processing procedure (e.g., in [40]), and also designing algorithms solely based on local search techniques such as those in [1, 14, 15, 34, 36, 37]. Recently, there is an increasing research interest in problem landscape analysis for MOPs. Such an interest includes analysing the problem characteristics (e.g., the connection among Pareto optimal solutions [18, 33,

*Corresponding author

GECCO '23, July 15–19, 2023, Lisbon, Portugal

ACM ISBN 979-8-4007-0119-1/23/07…$15.00
https://doi.org/10.1145/3583131.3590447

38, 41]), understanding the relation between the problem geometry and search dynamics (e.g., the effect of the problem features on algorithm performance [11, 32]), and developing automate algorithm selection based on the problem features (e.g., predicting the performance of search algorithms and selecting right algorithm components [6, 9, 32, 35].

In contrast to the above studies that develop hybrid/local search algorithms or analyse problem landscape, we, in this work, present an interesting behaviour of popular MOEAs on combinatorial MOPs (compared to on continuous ones). Our observations echo the importance of existing studies – the failure of well-established MOEAs on combinatorial MOPs suggests the need of developing effective search algorithms, particularly by designing hybrid/customised algorithms based on problem features.

The rest of the paper is organised as follows. Section II introduces the preliminaries of multi-objective optimisation and the MOEAs to be investigated (i.e., NSGA-II [12], SMS-EMOA [5] and MOEA/D [43]). Section III gives the considered combinatorial and continuous MOPs. Section IV describes the metric that is used to measure the distance between populations. Section V presents the experimental results. Lastly, Section VI concludes the paper.

2 MULTI-OBJECTIVE OPTIMISATION

2.1 Definitions

Without loss of generality, let us consider a minimisation problem with n decision variables and m objective functions $f : X \rightarrow Z$, with $X \subseteq \mathbb{R}^n$ and $Z \subseteq \mathbb{R}^m$. The objective functions map a point $x \in X$ in the decision space (i.e., search space) to a vector $z \in Z$ in the objective space such that $z = f(x) = (f_1(x), ..., f_m(x))$. Given two objective vectors $z, z' \in Z$, z is said to dominate z' (denoted by $z \prec z'$) iff for all $i \in \{1, ..., m\}$, $z_i \leq z'_i$, and there is a $j \in \{1, ..., m\}$ such that $z_j < z'_j$. Likewise, given two solutions $x, x' \in X$, x is said to dominate x' iff $f(x)$ dominates $f(x')$. An objective vector z is called Pareto optimal if there does not exist any $z' \in Z$ such that $z' \prec z$. A solution $x \in X$ is Pareto optimal if $f(x)$ is Pareto optimal. The set of Pareto optimal solutions is called the Pareto optimal set, and its mapping in the objective space is called the Pareto front. Usually, the size of an MOP's Pareto optimal set is prohibitively large, and it is not practical to find all Pareto optimal solutions. Therefore, the goal of a multi-objective optimiser is to find a representative solution set (with a much small size) of the Pareto set so that the decision-maker can choose their most preferred solution from the set.

2.2 MOEAs Investigated

Evolutionary algorithms have demonstrated their effectiveness in dealing with MOPs. Their nature of population-based search can approximate the whole Pareto front within one execution. In general, there are three classes of multi-objective evolutionary algorithms (MOEAs) based on their selection mechanisms: the Pareto-based, the indicator-based and the decomposition-based [17].

Non-dominated Sorting Genetic Algorithm II (NSGA-II) [12]. As one of the most widely used MOEAs, NSGA-II belongs to the Pareto-based algorithm class, characterised by non-dominated sorting [20] and crowding distance [12] procedures. In the selection process of NSGA-II, candidate solutions are divided into different non-dominated fronts. Solutions at lower fronts are regarded better than solutions at higher fronts. If solutions are at the same front, their crowding distance kicks in; the solutions with larger crowding distance are preferred, so as to maintain the diversity of solutions in the population.

***S* Metric Selection Evolutionary Multiobjective Optimisation Algorithm (SMS-EMOA)** [5]. SMS-EMOA is a representative algorithm based on a quality indicator. It follows the non-dominated sorting procedure of NSGA-II, but instead of crowding distance, it uses the hypervolume indicator[1] [44] to distinguish between solutions at the same nondominated front. That is, SMS-EMOA calculates the contribution of the hypervolume that each solution makes to the population, and removes solutions with the least contribution one by one. Since the hypervolume has desirable theoretical properties (e.g., strict monotonicity with respect to set dominance) [21] and can reflect both convergence and diversity of a solution set, SMS-EMOA has been commonly used to tackle a range of MOPs.

Multiobjective Evolutionary Algorithm Based on Decomposition (MOEA/D) [43]. MOEA/D is arguably the most popular decomposition-based algorithm. It converts a multiobjective problem into a number of scalar optimisation subproblems by a set of well-distributed weight vectors and an achievement scalarising function. The distribution of the weight vectors is intentionally used to maintain the diversity of solutions in the population; though it may not always work [24], there is some way to mitigate it through adapting the weight vectors during the search [30]. In MOEA/D, neighbouring sub-problems work collaboratively, e.g., sharing the best solutions and performing recombination. This is a major difference from the other two classes of MOEAs. Some recent studies have shown that MOEA/D has very different behaviours from other algorithms [35]. In our study, the well-establish ed Tchebycheff scalarising function was used and the reference point is updated (with the best value found on each objective) during the search.

3 OPTIMISATION PROBLEMS CONSIDERED

3.1 Multi-Objective NK-Landscape Problem (NK-Landscape)

NK-Landscape [3] is a commonly used problem in multi-objective optimisation due to its important feature that the ruggedness level of its problem landscape is tunable [42]. In the NK-landscape problem, N represents the number of bits (i.e., decision variables), and K represents the number of its neighbouring bits that interacts with a bit. Specifically, for each objective f_j, each bit x_i is associated with a contribution function $c_{ij}(x_i, x_{k_{ij1}}, \ldots, x_{k_{ijK}})$, where index k_{ij1} denotes the first bit of its loci k_{ij} for the bit x_i on the jth objective, and so on. The problem is defined as follows:

$$\max f_j(x) = \frac{1}{N} \sum_{i=1}^{N} c_{ij}(x_i, x_{k_{ij1}}, x_{k_{ij2}}, ..., x_{k_{ijK}}), \quad j = 1, ..., m$$

[1] Hypervolume is a quality indicator that measures the quality of a solution set by calculating the volume enclosed by the set and a specified reference point.

Following the common practice [2, 10], c_{ij} was set to be a fitness contribution drawn randomly from $[0, 1)$ for every possible combination $\{0, 1\}^{(K+1)}$ of the loci k_{ij}.

3.2 Multi-Objective Travelling Salesman Problem (TSP)

TSP [8] is a well-known combinatorial optimisation problem and it can be stated as follows. Given a number of cities and the cost of travel between each pair of them, the aim of the problem is to find the route that has the lowest traveling cost for visiting all the cities exactly once and returning to the starting point. The multi-objective TSP is an extended version of the standard TSP, in which there are multiple costs of travel between each pair of cities. Formally, the problem can be defined as: given a network $L = (V, C)$, where $V = \{v_1, \ldots, v_n\}$ is a set of n nodes and $C = \{C_j : j \in \{1, \ldots, m\}\}$ is a set of m cost matrices between nodes ($C_j : V \times V$), the task is to find the Pareto optimal set of Hamiltonian cycles that minimise each of the m cost objectives. In our experiments, the m matrices are independent, generated by assigning each pair of cities with a number randomly drawn from $[0, 1)$ [8].

3.3 Continuous Problems

We consider two commonly used continuous problem suites, the DTLZ [13] and WFG [22] suites. The DTLZ suite consists of seven problems (DTLZ1–DTLZ7) and the WFG suite consists of nine problems (WFG1–WFG9). These 17 problems can reflect a wide range of problem characteristics, such as linear, convex, concave, disconnected, biased and/or degenerated Pareto fronts, multi-modality, and linkage/non-separability among variables.

4 DISTANCE METRIC

Since we want to know the distance between populations, a distance metric is needed. Before defining the distance between populations, we need to define the distance between solutions. Such distance relies on the encoding for the considered problem (i.e., representation of the problem's variables in the evolutionary search).

4.1 Distance between Solutions

4.1.1 For NK-landscape. As for the NK-landscape problem in which the binary representation is used, the distance between two solutions x and y can be defined by Hamming distance straightforward:

$$\delta(x, y) = \frac{\sum_{i=1}^{n} |x_i - y_i|}{n} \tag{1}$$

where n denotes the number of variables and $x_i, y_i \in \{0, 1\}$. As can be seen from Eq. (1), the distance metric here is a normalised one, meaning that the minimum and maximum are 0 and 1, respectively. This can reflect how far solutions are located, relative to the possible maximum distance in the search space. For example, for two four-bit solutions 0000 and 0011, their distance is 0.5 according to Eq. (1), meaning that they are half way apart in the search space.

4.1.2 For TSP. TSP is a special permutation problem where the connection (i.e., edge) between variables (i.e., vertexes) matters, rather than the positions of variables. For example, solutions with the representations 12345 and 54321 are essentially the same in TSP, despite the fact that they have different positions of the variables in the representation. Here we define the distance between two solutions of TSP as follows:

$$\delta(x, y) = \frac{n - |E_x \bigcap E_y|}{n} \tag{2}$$

where E_x and E_y are sets of the edges of solutions x and y, respectively, and n denotes the number of edges (i.e., the number of cities). As can be seen, the distance of solutions 12345 and 54321 is 0; the distance of solutions 12345 and 12354 is 0.4; and the distance of solutions 12345 and 13524 is 1.

Note that the distance metric in [39] for general permutation problem may not be suitable here since it focuses on the positions of variables rather than their connections; according to the definition in [39], solutions 12345 and 54321 are very different solutions.

4.1.3 For Continuous Problems. Similar to in the binary representation, the distance between two solutions x and y in the real-number representation can be defined as:

$$\delta(x, y) = \frac{\sum_{i=1}^{n} \frac{|x_i - y_i|}{U_i - L_i}}{n} \tag{3}$$

where $x_i, y_i \in [L_i, U_i]$, and L_i and U_i are the lower and upper bounds on the ith objective, respectively. Apparently, $\delta(x, y) \in [0, 1]$, where $\delta(x, y) = 0$ means the two solutions being duplicate and $\delta(x, y) = 1$ means the two solutions being the most distant in the search space.

4.2 Distance between Solution Sets

After defining the distance between two solutions, we are in a position to define the distance between two populations (i.e., solution sets). We hope that, like the distance between solutions, the distance metric of sets can return 0 if the two sets are completely overlapping and return 1 if the sets are the possible farthest distance in the space. Formally, let $P = \{p_1, p_2, ..., p_{N_P}\}$ and $Q = \{q_1, q_2, ..., q_{N_Q}\}$ be two solution sets, where N_P and N_Q be their cardinalities, and then their distance can be defined as:

$$\Delta(P, Q) = \frac{\sum_{i=1}^{N_P} \delta(p_i, Q) + \sum_{i=1}^{N_Q} \delta(q_i, P)}{N_P + N_Q}, \tag{4}$$

where the distance of solution p_i to set Q is

$$\delta(p_i, Q) = \min_{q \in Q} \delta(p_i, q).$$

As can be seen from Eq. (4), the distance between two sets is an average minimum distance of each solution in any set to solutions in the other set. It reflects how close two sets are. If there is only one solution in each set, it reduces to the distance between two solutions. For example, in the binary representation, if one set $P = \{0000\}$ and another set $Q = \{0011\}$, then $\Delta(P, Q) = 1/2$. If now we add one solution 0001 in P (i.e., $P = \{0000, 0001\}$ and $Q = \{0011\}$), then the distance of the two sets reduces to $1/3$ ($\Delta(P, Q) = (1/2 + 1/4 + 1/4)/3 = 1/3$). This reduction is caused by the fact that there is now one solution in P (i.e., 0001) close to the solution 0011 of Q. If next we add one solution 0000 in Q (i.e., $P = \{0000, 0001\}$ and $Q = \{0000, 0011\}$), then the distance of the two sets further reduces to $1/8$ ($\Delta(P, Q) = (0 + 1/4 + 0 + 1/4)/4 = 1/8$) because the distance for the solution 0000 in each set becomes zero to the other set.

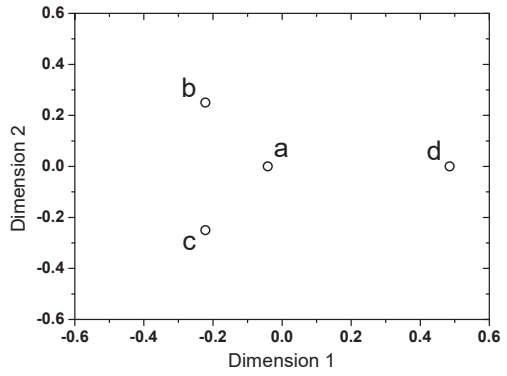
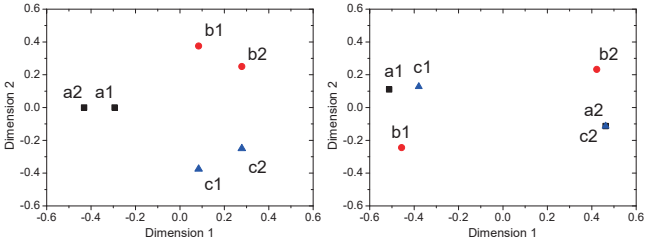


Figure 1: An illustration of four solutions with four-bit binary representations obtained by multi-dimensional scaling (MDS) [26] based on their pairwise distances. The four points are a=0000, b=0001, c=0010, and d=1100. According to Eq. (1), the pairwise distances between them are $\delta(a,b) = \delta(a,c) = 1/4$, $\delta(a,d) = \delta(b,c) = 1/2$, and $\delta(b,d) = \delta(c,d) = 3/4$.



(a) Three distant solution sets (b) Three close solution sets

Figure 2: An illustration of MDS for the case (a) that solution sets are located far away and the case (b) that solution sets are close. In sub-figure (a), three sets are A=$\{a_1, a_2\}$, B=$\{b_1, b_2\}$, C=$\{c_1, c_2\}$, where a_1=00000000, a_2=10000000, b_1=01111000, b_2=01111100, c_1=00001111, and c_2=00011111. In sub-figure (b), three sets are A=$\{a_1, a_2\}$, B=$\{b_1, b_2\}$, C=$\{c_1, c_2\}$, where a_1=00000000, a_2=11111111, b_1=10000000, b_2=01111111, c_1=01000000, and c_2=11111111.

4.3 Multidimensional Scaling (MDS)

To help visually understand results, we also consider a visualisation tool, Multidimensional Scaling (MDS) [26] to show locations of different populations in a 2D graph. MDS translates the pairwise distances between solutions into a 2D abstract Cartesian space. Specifically, given a distance matrix with the distances between each pair of solutions, MDS places all solutions into a 2D space such that the pairwise distances between the solutions are preserved as well as possible.

Figure 1 gives an example of MDS for four solutions of four-bit binary representations, a=0000, b=0001, c=0010, and d=1100. According to Eq. (1), the pairwise distances between them are $\delta(a,b) = \delta(a,c) = 1/4$, $\delta(a,d) = \delta(b,c) = 1/2$, and $\delta(b,d) = \delta(c,d) = 3/4$. As can be seen from the figure, MDS can well preserve pairwise distances for most solutions. For example, the distance of a to b is kept equivalent to that of a to c; the distance of d to b is equivalent to that of d to c; the distance of a to d is approximately equivalent to that of b to c. Of course, not all the distance relations can be perfectly preserved [26]. For example, the distance between c and d should equal the sum of the distance between c and a and the distance between a and d (as $\delta(c,d) = 3/4$, $\delta(c,a) = 1/4$ and $\delta(a,d) = 1/2$); however, in Figure 1 the distance between c and d is smaller than the sum of the distance of c to a and the distance of a to d.

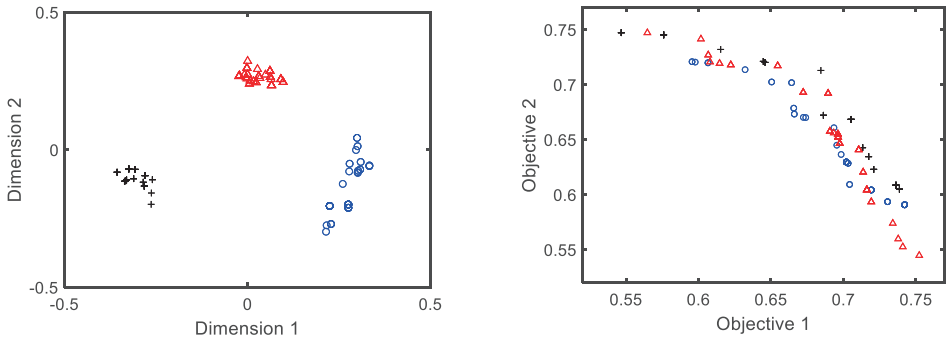
We can use MDS to reflect relative location of different sets. That is, we calculate the pairwise distance between all solutions of all the sets according to the distance metric (e.g., Eq. (1) for the binary representation), and the use MDS to map them into the 2D space. With different colours/shapes to represent different solution sets, we then know their relative location in the space. Figure 2 gives such an example, where there are two very different groups of solution sets mapped into the 2D space by MDS. In the first group, the three solution sets are distant, while their own two solutions are close. As can be seen in Figure 2(a), MDS can capture this. In the second group the three sets are close (i.e., for any solution in one set, there is always another solution in other sets nearby), while the two solutions in each set are far away to each other. Figure 2(b) well captures that. Note that since solutions a_2 and c_2 are duplicate, their mapped points are overlapping in the figure.

5 RESULTS

5.1 Experimental Design

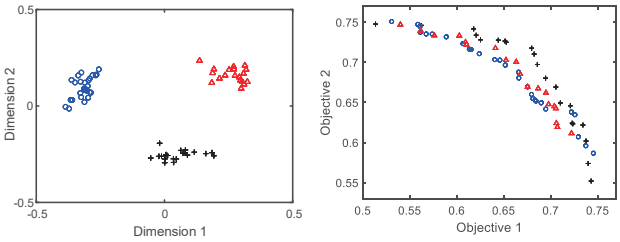
As described previously, we consider several commonly-used MOPs in both combinatorial and continuous optimisation, with normal setting of their parameters. Regarding the problem size (i.e., number of decision variables), for the continuous problem suites DTLZ and WFG, the recommended settings in their original studies [13, 22] were used. For NK-landscape, following the practice [2], N and K were set to 50 and 10, respectively. For TSP, N (the number of cities) was set to 100 according to [34, 38]. In the following experiments, we will first consider the bi-objective case with the above settings (Section 5.2 and Section 5.3). Then, in Section 5.4 we will study the effect of the problems with different problem sizes and numbers of objectives, as well as different levels of ruggedness.

Three well-known algorithms, NSGA-II [12], SMS-EMOA [5] and MOEA/D [43], are considered, representing the three types of Pareto-, indicator- and decomposition-based MOEAs, respectively. In MOEA/D, the Tchebycheff scalarising function was used. All the parameters of the algorithms were configured as the same as in their original papers. Common setting of the parameters in evolutionary search was used. The population size was set to 100. Regarding variation operators, for NK-landscape, according to [32] the single-point crossover with a rate of 0.9 and the bit-flip mutation with a rate of $1/n$ were used, where n denotes the number of variables. For TSP, the ordered crossover with a rate of 1.0 and the insertion mutation with a rate of $1/n$ were used following the practice in [19, 29]. For continuous MOPs, the simulated binary crossover with a rate of 1.0 and polynomial mutation with a rate of $1/n$ were considered. The distribution parameters for both crossover and mutation were set to 20 [43].



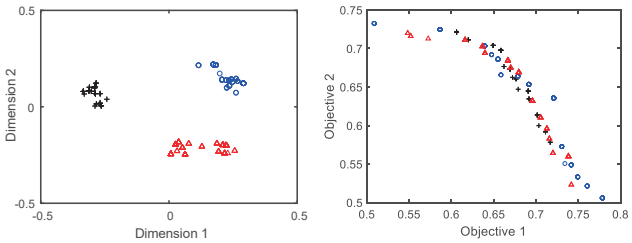
(a) Solution distribution in the decision space (mapped through MDS) (b) Their images in the objective space

Figure 3: Three populations (represented by plus, triangle and circle) obtained by NSGA-II under three random executions on the NK-Landscape problem ($N = 50, K = 10$). (a) Solutions mapped into the 2D space by MDS (multi-dimensional scaling) (b) Their images in the objective space.



(a) Decision space (mapped through MDS) (b) Objective space

Figure 4: Three populations (represented by plus, triangle and circle) obtained by SMS-EMOA under three random executions on the NK-Landscape problem ($N = 50, K = 10$). (a) Solutions mapped into the 2D space by MDS (b) Their images in the objective space.



(a) Decision space (mapped through MDS) (b) Objective space

Figure 5: Three populations (represented by plus, triangle and circle) obtained by MOEA/D under three random executions on the NK-Landscape problem ($N = 50, K = 10$). (a) Solutions mapped into the 2D space by MDS (b) Their images in the objective space.

Since we want to know the (relative) locations of the population after the algorithm stagnates, we need to define when we can call the population stagnant. To do so, we introduce an unbounded archive to store all unique nondominated solutions ever generated during the evolutionary process. Clearly, if the archive is unchanged for a while during the search, which means that there is no new nondominated solution generated, then the population may stagnate. Here, we set "for a while" to be 100 generations (i.e., after 10,000 new solutions generated). Yet for continuous MOPs, since the objective values are typically of infinite precision, an MOEA can still generate new nondominated solutions even if its search is trapped into local optima, in which case an unbounded archive based on Pareto dominance is not feasible [28]. To deal with that, we add the ϵ-dominance (with $\epsilon = 0.01$) as well [27]. That is to say, the archive accepts a new solution only if either the new solution is not ϵ-dominated by any solution in the archive, or the new solution Pareto dominates at least one solution in the archive (i.e., using the ϵ-Pareto archiving in [27]).

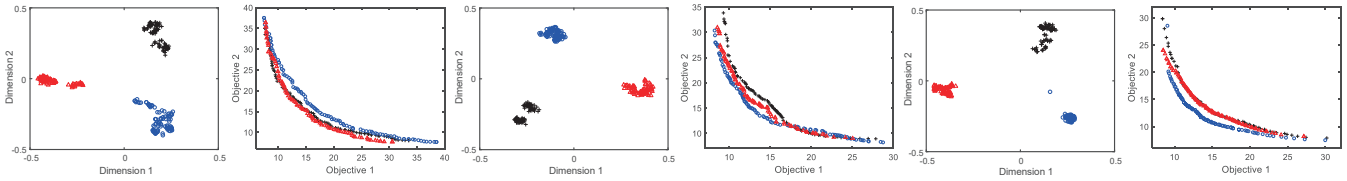
To calculate the distance metric of populations obtained by different executions of an MOEA, we execute each algorithm 10 times. As the distance metric is a pairwise metric (see Eq. (4) in Section 4.2), we consider all pairwise combinations of the 10 populations (i.e., $\binom{10}{2} = 45$ combinations) and calculate their mean and standard deviation.

5.2 On Combinatorial Problems

Table 1 gives the distance results on all the tested problems. Let us first consider the results on the NK-landscape problem. As seen in the table, the distance between the populations obtained by different executions of NSGA-II in the search space is 0.4679. This indicates that there are nearly half genes different between the closest solutions from different populations (see Eq. (4)). To help visually understand the distribution of the populations, we plot three populations obtained by NSGA-II under three random executions through mapping them into 2D space by the multi-dimensional scaling (MDS), along with their distribution in the objective space in Figure 3. It is clear that the three populations are agglomerated into three clusters in the space, despite the fact that their images in the objective space are mixed up together. This phenomenon also happens to SMS-EMOA and MOEA/D, as shown in Figures 4 and 5, where the same pattern can be spotted. This indicates that MOEAs stagnate in very different areas in the search space in different executions on the NK-landscape problem.

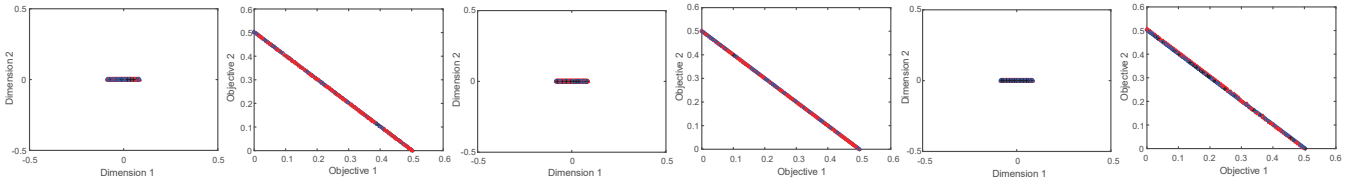
Table 1: Statistical results of the Distance Metric (mean and SD) between populations in the search space, obtained by NSGA-II, SMS-EMOA and MOEA/D under ten executions on all the considered MOPs, where the distance metric value 0 represents that the populations are overlapping completely and 1 represents that the populations are located the possible farthest distance to each other in the search space (see Section 4.2). For NK-landscape and TSP, there is one problem instance considered on each problem; for the DTLZ/WFG suite, there are multiple problem instances (DTLZ1–DTLZ7/WFG1–WFG9) and the results are the statistical results (mean and SD) on all the problem instances.

Problem	NSGA-II	SMS-EMOA	MOEA/D
NK-landscape	4.679E-01(6.6E-02)	4.874E-01(4.1E-02)	4.781E-01(4.9E-02)
TSP	7.986E-01(1.7E-02)	7.799E-01(2.0E-02)	7.328E-01(2.7E-02)
DTLZ	2.411E-03(5.4E-04)	7.936E-04(2.6E-04)	1.803E-04(4.4E-05)
WFG	2.205E-03(5.1E-04)	2.393E-03(1.9E-04)	9.738E-04(1.8E-04)



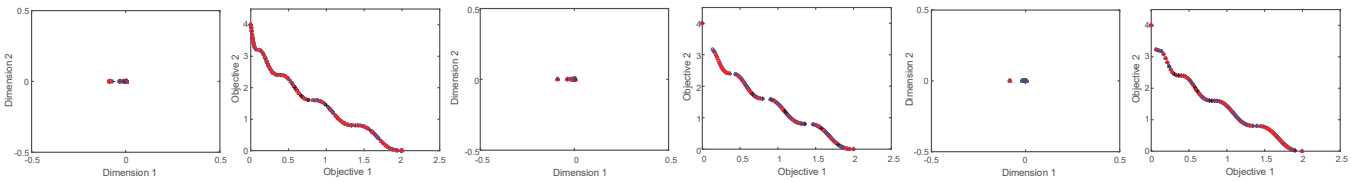
(a) Decision space (NSGA-II) (b) Objective space (NSGA-II) (c) Decision space (SMS-EMOA) (d) Objective space (SMS-EMOA) (e) Decision space (MOEA/D) (f) Objective space (MOEA/D)

Figure 6: Three populations (represented by plus, triangle, and circle) obtained by NSGA-II, SMS-EMOA and MOEA/D under three random executions on TSP ($N = 100$). Sub-figures (a), (c), (e) show the solutions (obtained by the specific MOEA) mapped into the 2D space by MDS, and sub-figures (b), (d), (f) show their corresponding images in the objective space.



(a) Decision space (NSGA-II) (b) Objective space (NSGA-II) (c) Decision space (SMS-EMOA) (d) Objective space (SMS-EMOA) (e) Decision space (MOEA/D) (f) Objective space (MOEA/D)

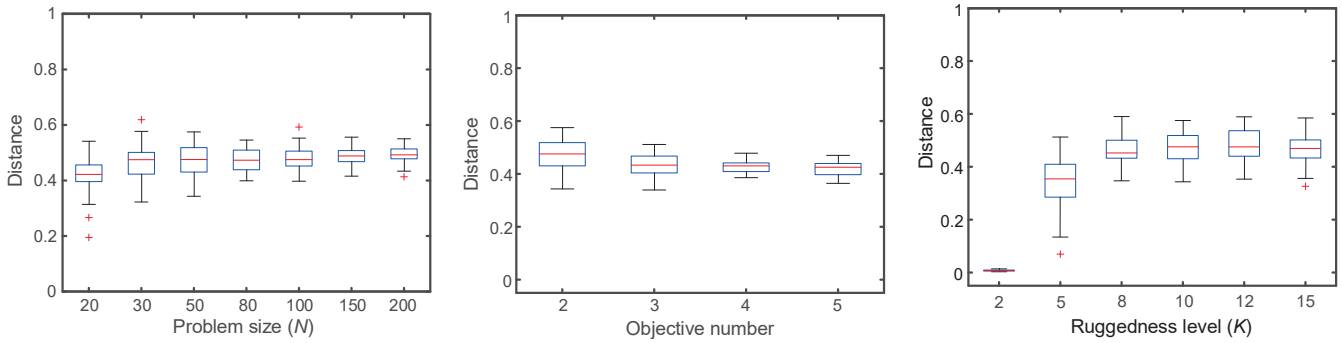
Figure 7: Three populations (represented by plus, triangle, and circle) obtained by NSGA-II, SMS-EMOA and MOEA/D under three random executions on the bi-objective DTLZ1. Sub-figures (a), (c), (e) show the solutions (obtained by the specific MOEA) mapped into the 2D space by MDS, and sub-figures (b), (d), (f) show their corresponding images in the objective space.



(a) Decision space (NSGA-II) (b) Objective space (NSGA-II) (c) Decision space (SMS-EMOA) (d) Objective space (SMS-EMOA) (e) Decision space (MOEA/D) (f) Objective space (MOEA/D)

Figure 8: Three populations (represented by plus, triangle, and circle) obtained by NSGA-II, SMS-EMOA and MOEA/D under three random executions on the bi-objective WFG1. Sub-figures (a), (c), (e) show the solutions (obtained by the specific MOEA) mapped into the 2D space by MDS, and sub-figures (b), (d), (f) show their corresponding images in the objective space.

Now consider the other combinatorial problem TSP. The results in Table 1 and Figure 6 show that populations obtained by different executions of the MOEAs are even farther apart to each other, though their images in the objective space are very close. As seen in Figure 6, the three populations are located almost on three "corners" in the search space. The closest solutions from different populations have the distance metric of over 0.7 on average (i.e., more than 70% genes are different) for all the algorithms (Table 1).



(a) Problem size (b) Number of objectives (c) Ruggedness level

Figure 9: Effect of different problem parameters to the distance between populations obtained by different executions of NSGA-II on the NK-landscape problem.

5.3 On Continuous Problems

The results on the continuous problems are very opposite. Table 1 also shows the distance results on the continuous problem suites DTLZ and WFG, where the mean and SD of the distance results on all the problem instances (i.e., DTLZ1–DTLZ7 and WFG1–WFG9) are given. As can be seen from the table, the distance between the populations on both the DTLZ and WFG suites is less than 0.0025 for all the MOEAs, meaning that their populations are located very closely. This can also be confirmed by Figures 7 and 8 on the problem instances of DTLZ1 and WFG1, where the populations obtained by three random executions of NSGA-II, SMS-EMOA or MOEA/D are overlapping in both decision space and objective space. This indicates that the algorithm stops with its solutions in the same area every time.

5.4 With Different Problem Difficulty

We have seen in the last two sections that in contrast to on continuous MOPs, on combinatorial MOPs populations obtained by different executions of an MOEA tend to be in different areas in the search space. Yet, all the experiments were conducted with a fixed setting of the problem parameters; for example, in NK-landscape, the problem size (i.e., N), the number of objectives, and the level of ruggedness (i.e., K) were set to 50, 2, and 10, respectively. One may wonder how the setting of problem parameters affects the behaviour of MOEAs. In this section, we aim to investigate this issue. In particular, we consider the algorithm NSGA-II on the NK-landscape problem, to see how different problem parameter settings affect the distance between the populations obtained by its different executions. Similar results can be observed in the other MOEAs.

5.4.1 Problem Size. We consider the problem size (i.e., number of variables) to be 20, 30, 50, 80, 100, 150 and 200. Figure 9(a) plots the trajectory of the average distance between the populations obtained by 10 executions of NSGA-II under NK-landscape with these problem sizes. As can be seen from the figure, the increase of the problem size (i.e., search space) does not affect the behaviour of NSGA-II significantly, though there is a slightly increasing tendency. The reason for this is because the distance is a normalised metric $\in [0, 1]$. Even if the absolute distance between the populations increases, the distance, relative to the increasing search space, stays similar. This indicates that the problem size may not be a crucial factor that affects the behaviour of MOEAs of stagnating in very different areas in the search space.

5.4.2 Number of Objectives. Next, we consider different numbers of objectives. Figure 9(b) plots the trajectory of the average distance between the populations obtained by 10 executions of NSGA-II on NK-landscape with 2, 3, 4, and 5 objectives. As seen in the figure, the number of objectives has little to do with the distance between the populations when NSGA-II stagnates, with the distance values for four different numbers of objectives all being around 0.4 and 0.5 (i.e., nearly half of the genes of the closest solutions from different populations being different).

5.4.3 Ruggedness. Lastly, let us consider different levels of ruggedness of the problem landscape. In NK-landscape, the parameter K is the one to determine that, with a larger K meaning that the problem landscape is more rugged. Here, we set $K = 2, 5, 8, 10, 12, 15$. As can be seen from Figure 9(c), ruggedness level of the problem landscape significantly affects the location of the populations obtained by different executions of NSGA-II. When the landscape is smooth (i.e., $K = 2$), the distance is close to zero, which means that the populations of different NSGA-II's executions end up in the same area in the space. Since we stop the algorithm only when it cannot generate new nondominated solutions for quite a while, that area may be the "ridge" in which the problem's optimal solutions are located. However, with the increase of the problem's ruggedness level, the populations are getting farther. Interestingly, when K reaches 8 or above, there is no much difference between the populations' distance. One possible explanation for this is that when $K = 8$, the algorithm is already completely stuck in local optima; so increasing the ruggedness level on those local optima will not affect the location of the populations.

To sum up, the level of ruggedness of the problem landscape is a crucial factor that affects the location of the populations. When the problem's ruggedness level reaches a certain threshold, MOEAs completely fail to jump out of local optima.

6 CONCLUSIONS

In this paper, we show that on combinatorial MOPs the search in every execution of an MOEA tends to stagnate in a different area in the search space. It is known that local search methods may result in "clusters" of their solutions on combinatorial MOPs [38]. Here, we present that this also happens to those population-based global search methods – MOEAs. In such MOEAs, populations obtained by different executions are never overlapping, and in fact are located far away apart to each other. This indicates the ineffectiveness of the diversity induced in state-of-the-art algorithms in guiding the search to explore different regions of the search space, and thus suggests the need of developing more effective global search methods to tackle combinatorial MOPs. Having that said, the conclusion drawn here is based on "normal setting" of MOEAs, e.g., an elitist selection process with a normal-size population; it may change if, for example, a non-elitist selection process with a big population is considered, which recently has been found to behave differently on combinatorial MOPs [31].

On the other hand, this finding may also imply some potential way of improving MOEAs' performance on combinatorial MOPs. For example, a straightforward idea is to maintain multiple populations and let them evolve independently during the evolutionary process, and when they stagnate, exchange their information for making the search jump out of local optima.

Another subsequent work is to investigate MOEAs on more and diverse combinatorial and continuous problems, particularly with practical features. In this paper, we considered commonly used synthetic MOPs, where the problem instances are randomly generated (e.g., the contribution matrices in NK-landscape and the cost matrices in TSP). However, considering other types of instance generation or real-world instances may result in different observations.

REFERENCES

[1] Hernan Aguirre and Kiyoshi Tanaka. 2005. Random bit climbers on multiobjective MNK-landscapes: effects of memory and population climbing. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 88, 1 (2005), 334–345.

[2] Hernán Aguirre and Kiyoshi Tanaka. 2007. Working principles, behavior, and performance of MOEAs on MNK-landscapes. *European Journal of Operational Research* 181, 3 (2007), 1670–1690.

[3] Hernán E Aguirre and Kiyoshi Tanaka. 2004. Effects of elitism and population climbing on multiobjective MNK-landscapes. In *IEEE Congress on Evolutionary Computation*, Vol. 1. 449–456.

[4] Thomas Back. 1996. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press.

[5] Nicola Beume, Boris Naujoks, and Michael Emmerich. 2007. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research* 181, 3 (2007), 1653–1669.

[6] Leonardo CT Bezerra, Manuel López-Ibánez, and Thomas Stützle. 2014. Automatic design of evolutionary algorithms for multi-objective combinatorial optimization. In *The 13th International Conference on Parallel Problem Solving from Nature*. Springer, 508–517.

[7] Aymeric Blot, Marie-Éléonore Kessaci, and Laetitia Jourdan. 2018. Survey and unification of local search techniques in metaheuristics for multi-objective combinatorial optimisation. *Journal of Heuristics* 24, 6 (2018), 853–877.

[8] David W. Corne and Joshua D. Knowles. 2007. Techniques for highly multiobjective optimisation: some nondominated points are better than others. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation Conference (GECCO)*. 773–780.

[9] Raphaël Cosson, Bilel Derbel, Arnaud Liefooghe, Sébastien Verel, Hernan Aguirre, Qingfu Zhang, and Kiyoshi Tanaka. 2022. Cost-vs-accuracy of sampling in multi-objective combinatorial exploratory landscape analysis. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 493–501.

[10] Fabio Daolio, Arnaud Liefooghe, Sébastien Verel, Hernán Aguirre, and Kiyoshi Tanaka. 2015. Global vs local search on multi-objective NK-landscapes: contrasting the impact of problem features. In *proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. 369–376.

[11] Fabio Daolio, Arnaud Liefooghe, Sébastien Verel, Hernán Aguirre, and Kiyoshi Tanaka. 2017. Problem features versus algorithm performance on rugged multiobjective combinatorial fitness landscapes. *Evolutionary computation* 25, 4 (2017), 555–585.

[12] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197.

[13] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. 2005. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, Ajith Abraham, Lakhmi Jain, and Robert Goldberg (Eds.). Berlin, Germany: Springer, 105–145.

[14] Mădălina M Drugan and Dirk Thierens. 2012. Stochastic Pareto local search: Pareto neighbourhood exploration and perturbation strategies. *Journal of Heuristics* 18 (2012), 727–766.

[15] Jérémie Dubois-Lacoste, Manuel López-Ibáñez, and Thomas Stützle. 2015. Anytime Pareto local search. *European journal of operational research* 243, 2 (2015), 369–385.

[16] Matthias Ehrgott and Xavier Gandibleux. 2008. Hybrid metaheuristics for multiobjective combinatorial optimization. In *Hybrid metaheuristics*. Springer, 221–259.

[17] Michael TM Emmerich and André H Deutz. 2018. A tutorial on multiobjective optimization: Fundamentals and evolutionary methods. *Natural Computing* 17, 3 (2018), 585–609.

[18] Jonathan E Fieldsend and Khulood Alyahya. 2019. Visualising the landscape of multi-objective problems using local optima networks. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 1421–1429.

[19] David B Fogel. 1988. An evolutionary approach to the traveling salesman problem. *Biological Cybernetics* 60, 2 (1988), 139–144.

[20] D.E. Goldberg. 1989. *Genetic algorithms in search, optimization, and machine learning*. Addison-wesley.

[21] Andreia P Guerreiro, Carlos M Fonseca, and Luís Paquete. 2021. The Hypervolume Indicator: Computational Problems and Algorithms. *ACM Computing Surveys (CSUR)* 54, 6 (2021), 1–42.

[22] S. Huband, P. Hingston, L. Barone, and L. While. 2006. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation* 10, 5 (2006), 477–506.

[23] Hisao Ishibuchi and Tadahiko Murata. 1998. A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Trans. Syst., Man, Cybern., C: Appl. Rev.* 28, 3 (1998), 392–403.

[24] Hisao Ishibuchi, Yu Setoguchi, Hiroyuki Masuda, and Yusuke Nojima. 2017. Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes. *IEEE Transactions on Evolutionary Computation* 21, 2 (2017), 169–190.

[25] A. Jaszkiewicz. 2002. On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - a comparative experiment. *IEEE Transactions on Evolutionary Computation* 6, 4 (2002), 402–412.

[26] Joseph B Kruskal and Myron Wish. 1978. *Multidimensional scaling*. Number 11. Sage.

[27] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. 2002. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation* 10, 3 (2002), 263–282.

[28] Miqing Li, Manuel López-Ibáñez, and Xin Yao. 2023. Multi-objective archiving. *arXiv preprint arXiv:2303.09685* (2023).

[29] Miqing Li, Shengxiang Yang, and Xiaohui Liu. 2015. Bi-goal evolution for many-objective optimization problems. *Artificial Intelligence* 228 (2015), 45–65.

[30] Miqing Li and Xin Yao. 2020. What weights work for you? Adapting weights for any Pareto front shape in decomposition-based evolutionary multiobjective optimisation. *Evolutionary Computation* 28, 2 (2020), 227–253.

[31] Zimin Liang, Miqing Li, and Per Kristian Lehre. 2023. Non-elitist evolutionary multi-objective optimisation: Proof-of-principle results. In *Genetic and Evolutionary Computation Conference*. ACM.

[32] Arnaud Liefooghe, Fabio Daolio, Sébastien Verel, Bilel Derbel, Hernan Aguirre, and Kiyoshi Tanaka. 2019. Landscape-aware performance prediction for evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation* 24, 6 (2019), 1063–1077.

[33] Arnaud Liefooghe, Bilel Derbel, Sébastien Verel, Manuel López-Ibáñez, Hernan Aguirre, and Kiyoshi Tanaka. 2018. On Pareto local optimal solutions networks. In *International Conference on Parallel Problem Solving from Nature*. Springer, 232–244.

[34] Arnaud Liefooghe, Jérémie Humeau, Salma Mesmoudi, Laetitia Jourdan, and El-Ghazali Talbi. 2012. On dominance-based multiobjective local search: design, implementation and experimental analysis on scheduling and traveling salesman problems. *Journal of Heuristics* 18 (2012), 317–352.

[35] Arnaud Liefooghe, Sébastien Verel, Bilel Derbel, Hernan Aguirre, and Kiyoshi Tanaka. 2020. Dominance, indicator and decomposition based search for multi-objective QAP: landscape analysis and automated algorithm selection. In *International Conference on Parallel Problem Solving from Nature*. Springer, 33–47.

[36] Thibaut Lust and Jacques Teghem. 2010. Two-phase Pareto local search for the biobjective traveling salesman problem. *Journal of Heuristics* 16, 3 (2010), 475–510.

[37] Luis Paquete, Marco Chiarandini, and Thomas Stützle. 2004. Pareto local optimum sets in the biobjective traveling salesman problem: An experimental study. In *Metaheuristics for multiobjective optimisation*. Springer, 177–199.

[38] Luís Paquete and Thomas Stützle. 2009. Clusters of non-dominated solutions in multiobjective combinatorial optimization: An experimental analysis. In *Multi-objective Programming and Goal Programming*. Springer, 69–77.

[39] Tommaso Schiavinotto and Thomas Stützle. 2007. A review of metrics on permutations for search landscape analysis. *Computers & operations research* 34, 10 (2007), 3143–3153.

[40] El-Ghazali Talbi, Malek Rahoual, Mohamed Hakim Mabed, and Clarisse Dhaenens. 2001. A hybrid evolutionary approach for multicriteria optimization problems: Application to the flow shop. In *Evolutionary Multi-Criterion Optimization: First International Conference, EMO 2001 Zurich, Switzerland, March 7–9, 2001 Proceedings 1*. Springer, 416–428.

[41] Shoichiro Tanaka, Keiki Takadama, and Hiroyuki Sato. 2022. Impacts of Single-objective Landscapes on Multi-objective Optimization. In *2022 IEEE Congress on Evolutionary Computation (CEC)*. 1–8.

[42] Sébastien Verel, Arnaud Liefooghe, Laetitia Jourdan, and Clarisse Dhaenens. 2013. On the structure of multiobjective combinatorial search space: MNK-landscapes with correlated objectives. *European Journal of Operational Research* 227, 2 (2013), 331–342.

[43] Qingfu Zhang and Hui Li. 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation* 11, 6 (2007), 712–731.

[44] Eckart Zitzler and Lothar Thiele. 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation* 3, 4 (1999), 257–271.