Many-objective (Combinatorial) Optimization is Easy

Arnaud Liefooghe
arnaud.liefooghe@univ-lille.fr
Univ. Lille, CNRS, Inria, Centrale Lille,
UMR 9189 CRIStAL
F-59000 Lille, France

Manuel López-Ibáñez
manuel.lopez-ibanez@manchester.ac.uk
Alliance Manchester Business School,
University of Manchester
M13 9PL Manchester, UK

ABSTRACT

It is a common held assumption that problems with many objectives are harder to optimize than problems with two or three objectives. In this paper, we challenge this assumption and provide empirical evidence that increasing the number of objectives tends to reduce the difficulty of the landscape being optimized. Of course, increasing the number of objectives brings about other challenges, such as an increase in the computational effort of many operations, or the memory requirements for storing non-dominated solutions. More precisely, we consider a broad range of multi- and many-objective combinatorial benchmark problems, and we measure how the number of objectives impacts the dominance relation among solutions, the connectedness of the Pareto set, and the landscape multimodality in terms of local optimal solutions and sets. Our analysis shows the limit behavior of various landscape features when adding more objectives to a problem. Our conclusions do not contradict previous observations about the inability of Pareto-optimality to drive search, but we explain these observations from a different perspective. Our findings have important implications for the design and analysis of many-objective optimization algorithms.

CCS CONCEPTS

• **Theory of computation** → *Optimization with randomized search heuristics*; *Design and analysis of algorithms*; • **Applied computing** → *Multi-criterion optimization and decision-making*.

KEYWORDS

Many-objective optimization, search difficulty, local search, multi-modality, local optima, multi-objective nk-landscapes.

ACM Reference Format:
Arnaud Liefooghe and Manuel López-Ibáñez. 2023. Many-objective (Combinatorial) Optimization is Easy. In *Genetic and Evolutionary Computation Conference (GECCO '23), July 15–19, 2023, Lisbon, Portugal*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3583131.3590475

1 BACKGROUND AND MOTIVATIONS

In contrast to single-objective optimization, multi-objective optimization deals with problems where multiple, typically conflicting objectives must be optimized simultaneously. Under this scenario, optimality is often defined in terms of Pareto dominance, i.e., a solution dominates another if the former is better than the latter in all objectives and strictly better in at least one of them. A solution that is not dominated by any other feasible solution is called Pareto-optimal. The set of Pareto-optimal solutions is the Pareto set and their image in objective space is the Pareto front.

It is known than adding objectives to a single-objective optimization problem may actually help solving it [17, 20], and it has been proven that adding a new objective to an existing multi-objective problem may reduce the runtime required by an evolutionary algorithm to find the Pareto front [2]. Nevertheless, increasing the number of objectives is generally thought to make optimization problems harder to solve. In recent years, a large amount of research effort has been devoted to many-objective optimization problems, that is, problems with four or more conflicting objectives. A survey from 2015 [22] already listed 238 papers on this topic and this number has surely grown exponentially since then.

One of the main challenges in multi-objective optimization is that the size of the Pareto set may be exponential on the input size, even with two objectives [6], and the fraction of solutions that are Pareto-optimal generally increases sharply with the number of objectives [38]. Therefore, most multi-objective optimization methods only aim at producing a high-quality bounded-size approximation of the Pareto front [19]. Even within this aim, the runtime of many steps within multi-objective optimization algorithms is at least polynomial and often exponential on the number of objectives [15], which obviously increases the computational requirements of the algorithms. Nevertheless, this is not due to any intrinsic difficulty of the optimization landscape, but simply to the computational complexity of the algorithms available for performing such steps. In other words, we would not say that a problem has become easier if we improve the runtime of an optimizer by using more efficient implementations or better algorithms for such steps [13, 16] as long as the optimizer visits exactly the same sequence of solutions during its execution. In fact, the comparison of multi- and many-objective optimizers is often performed in terms of the number of evaluated solutions, and the runtime required by other algorithmic steps is given less attention.

Therefore, we wish to set aside the runtime of individual algorithmic steps and focus instead on the landscape features of many-objective combinatorial optimization problems [3] that influence the number of solutions (or neighborhood evaluations) required to reach an approximation of the Pareto front with certain quality. High-level landscape features that makes an optimization problem harder typically include its ruggedness, the connectedness of the Pareto set, the number of local optima that are not global optima, etc. From this particular point of view, we claim here that many-objective problems are generally easier than problems with two or

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

ACM ISBN 979-8-4007-0119-1/23/07…$15.00
https://doi.org/10.1145/3583131.3590475

three objectives, from the point of view of finding a bounded size set of Pareto-optimal solutions. As we will show empirically, for more than five objectives almost all solutions are Pareto-optimal and the Pareto set is connected, thus finding multiple Pareto-optimal solutions becomes trivial even with an uniform random sampling. The connectedness of the Pareto set persists even with highly non-linear landscapes, which would typically induce highly disconnected landscapes. Our observations here are consistent and complement the empirical observation that Pareto dominance is unable to drive the search in many-objective optimization [14, 33]. We show that the reason is that, from the point of view of Pareto dominance, the landscape is basically neutral and connected, and the difficulty lies elsewhere.

Of course, there are other characteristics that may make a many-objective optimization problem difficult to solve. On top of the increase in computational complexity, many-objective optimization algorithms aim to find a "well-distributed" set of solutions by optimizing (directly or indirectly) unary quality metrics, such as the hypervolume, and we make no claims (yet) about whether those metrics are easier or harder to optimize with additional objectives. We also make not claims about the difficulty in optimizing scalarization functions that convert a many-objective problem into multiple single-objective ones [32].

The paper is organized as follows. After giving the experimental setup of our analysis in Section 2, we investigate the correlation among objectives and the dominance relations among solutions in Section 3, the connectedness of the Pareto set in Section 4, and the multimodality in terms of local optimal solutions and sets in Section 5. We finally summarize our main findings and discuss further research in Section 6.

2 EXPERIMENTAL SETUP

This section covers the experimental setup of our analysis, including the considered benchmark problems and their setting.

2.1 ρmnk-Landscapes

We consider ρmnk-landscapes as a problem-independent model for generating multi- and many-objective multimodal problems [1, 36]. Candidate solutions are binary strings of size n, and the objective function vector $f = (f_1, \dots, f_i, \dots, f_m)$ is defined as $f\colon \{0,1\}^n \mapsto [0,1]^m$ such that each objective f_i is to be maximized, $i \in \{1,\dots,m\}$. As in well-established single-objective nk-landscapes [18], the objective value $f_i(x)$ of a solution x is an average value of the individual contributions associated with each variable x_j. Given objective f_i, $i \in \{1,\dots,m\}$, and variable x_j, $j \in \{1,\dots,n\}$, a component function $f_{ij}\colon \{0,1\}^{k+1} \mapsto [0,1]$ assigns a real-valued contribution for every combination of x_j and its k *variable interactions* $\{x_{j_1},\dots,x_{j_k}\}$. These f_{ij}-values are uniformly distributed in $[0,1]$. Thus, the individual contribution of a variable x_j depends on its value and on the values of $k < n$ other variables $\{x_{j_1},\dots,x_{j_k}\}$. The variable interactions, i.e., the k variables that influence the contribution of x_j, are set uniformly at random among the $(n-1)$ variables other than x_j [18]. By increasing the number of variable interactions k from 0 to $(n-1)$, problems can be gradually tuned from smooth to rugged. In ρmnk-landscapes, f_{ij}-values follow a multivariate uniform distribution of dimension m, defined by an $m \times m$ positive-definite symmetric covariance matrix (c_{pq}) such that $c_{pp} = 1$ and $c_{pq} = \rho$ for all $p, q \in \{1,\dots,m\}$ with $p \neq q$, where $\rho > \frac{-1}{m-1}$ defines the correlation among the objectives; see [36] for details. The positive (resp. negative) objective correlation ρ decreases (resp. increases) the degree of conflict between the objective function values. The correlation coefficient ρ is the same for each pair of objectives, and the variable interactions are the same for all the objectives. By construction, it is very unlikely that different solutions map to the same point in the objective space. Interestingly, ρmnk-landscapes exhibit different characteristics and different degrees of difficulty for multi-objective algorithms [3, 23]. The ρmnk-landscapes generator is available at the following URL: http://mocobench.sf.net.

2.2 Problems Setting

We generate 540 multi- and many-objective ρmnk-landscapes under the following settings:

- The problem size is $n = 14$, so that the solution space can be enumerated exhaustively.
- The problem non-linearity is $k \in \{1, 2, 4\}$, i.e. from relatively smooth to relatively rugged problems.
- The number of objectives is $m \in \{2, 3, 4, 5, 7, 10, 15, 20\}$; we also consider the single-objective case ($m = 1$) when relevant.
- The correlation among objectives is $\rho \in \left\{0, 0.95 \cdot \frac{-1}{m-1}\right\}$, i.e. from uncorrelated to highly conflicting objectives.

For each problem setting, 10 random instances are independently generated.

3 OBJECTIVE CORRELATION AND DOMINANCE

We start by analyzing how the number of objectives impacts their degree of conflict and the dominance relation between solutions.

3.1 Correlation among Objectives

Strongly correlated objectives do not add any difficulty to a problem, since optimizing for one objective will optimize the other. Intuitively, uncorrelated or negatively correlated objectives would be more challenging that positively correlated ones from the point of view of optimization. However, as we will show, the more we increase the number of objectives, the less conflicting they can be.

For the degree of conflict among the objectives, we consider the pairwise correlation between the objective values. More precisely, given a pair of objectives (f_i, f_j), we measure the Spearman rank correlation coefficient between f_i- and f_j-values, i.e. a non-parametric measure of rank correlation to account for potential non-linearities. The pairwise correlation ranges from -1.0 (highly conflicting objectives) to 1.0 (highly correlated objectives), passing through 0.0 (uncorrelated or independent objectives), and varies according to the degree of conflict between f_i and f_j. We then average the correlation coefficients over all pairs of objectives.

This is reported first over the entire solution space in Figure 1. The average pairwise correlation is given with respect to the number of objectives m (x-axis) and to the problem non-linearity k (in colors), for both uncorrelated (left) and conflicting objectives (right). As expected, the objective correlation is close to 0.0 for uncorrelated

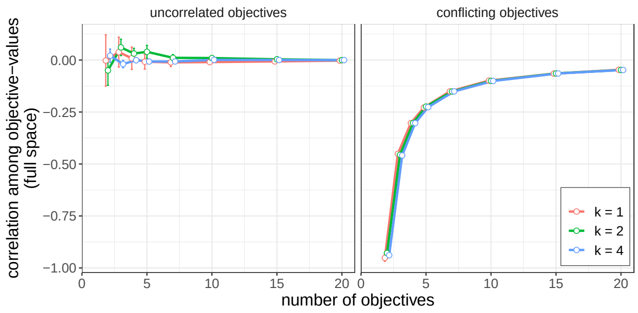

Figure 1: Average correlation among pairs of objectives (full space).

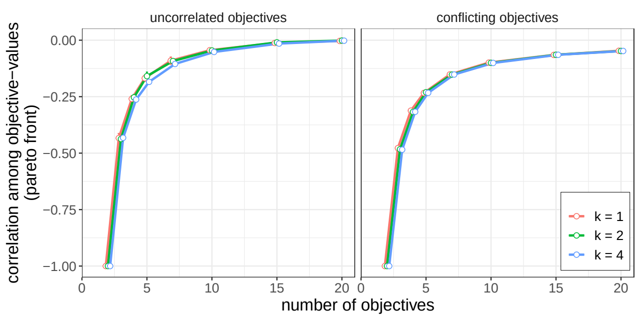

Figure 2: Average correlation among pairs of objectives (Pareto-optimal solutions only).

objectives, independently of m and k. For conflicting objectives, the curve follows the setting of $0.95 \cdot \frac{-1}{m-1}$ described in the problems setting of Section 2, and thus only depends on m. For more than two objectives, we cannot have all pairs of objectives strongly in conflict one another. Let us consider the example where two objectives have a correlation of -1.0. If we now add a third objective that conflicts with the first one, it will necessarily be positively correlated to the second one (and vice versa). More formally, for any multi-objective problem, if all pairwise correlations are the same, the correlation cannot be lower than $\frac{-1}{m-1}$, because the correlation matrix is a covariance matrix and must therefore be positive semi-definite [34]. Therefore, the correlation tends towards 0.0 as m increases. This explains why the pairwise correlation tends to increase on average with the number of objectives. By construction, the objective correlation of ρmnk-landscapes is the same for all pairs of objectives, which explains why the confidence interval is quite small.

Let us now comment on the average pairwise objective correlation for solutions from the Pareto front only, as shown in Figure 2. Because the solutions are mutually non-dominated, the objective correlation is -1.0 when $m = 2$: given two Pareto-optimal solutions, when one is better for f_1, it is necessarily worse for f_2. However, as the number of objectives increases, the average pairwise correlation also increases, and seems to follow a logarithmic growth tending towards 0.0. Although it is consistently higher for uncorrelated objectives, where it reaches a value of 0.0 for 15 objectives, it is just

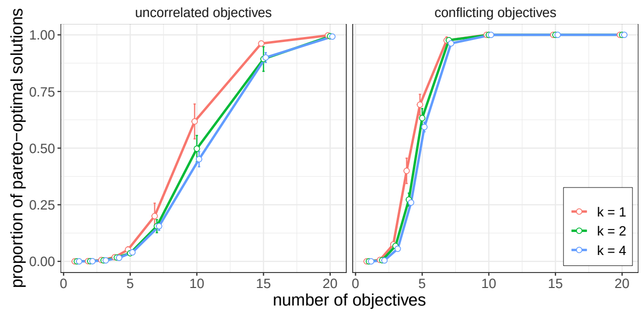

Figure 3: Proportion of solutions being Pareto-optimal.

slightly behind for conflicting objectives, and goes from -1.0 for $m = 2$ to about -0.05 for $m = 20$. Overall, whether considering all solutions or Pareto-optimal solutions only, the degree of conflict between the objectives seems to decrease with the number of objectives. For the solutions in the Pareto front, their correlation quickly converges towards zero when adding more objectives, even if the objectives are as mutually conflicting as possible.

Although we cannot claim at this stage that zero correlation in the Pareto front makes a problem easier or harder to solve, it is well-known that some algorithms behave better than others depending on the objective correlation; see, e.g., [28, 30]. Since the correlation seems to converge to zero regardless of the actual degree of conflict between the objectives and the value of k, this would suggest that some classes of algorithms may have an advantage over many-objective optimization problems.

3.2 Dominance Relations among Solutions

We now investigate the probability for a (random) solution to be Pareto-optimal depending on the number of objectives. We argue that a problem is intuitively easier if finding a Pareto-optimal solution by chance is more likely.

Figure 3 gives the proportional number of non-dominated solutions within the solution space. We observe that in both cases (uncorrelated and conflicting objectives), a handful of solutions are non-dominated when there are few objectives, while this increases substantially as the number of objectives grows. More specifically, for uncorrelated objectives, this proportion is below 5% for $m \leq 5$ objectives and reaches 100% for $m = 20$. For conflicting objectives, it already goes above 5% for $m \geq 3$, and reaches 100% for $m = 10$. This means that for $m \geq 20$ uncorrelated objectives or $m \geq 10$ conflicting objectives, all solutions are Pareto-optimal for the considered problems. Here as well, the problem non-linearity has little impact, although we note that there are slightly fewer Pareto-optimal solutions when k is larger.

In Figure 4, we complete our analysis by calculating, for each solution, how many other solutions dominate it, and we average over the whole solution space. In fact, this matches the strategy used by Fonseca and Fleming's MOGA [8] to rank solutions from a population. The average proportion of dominating points per solution is about half for 1 objective, by construction of nk-landscapes [18]. It then follows a logarithmic decrease with respect to the number of objectives, to the point where there are almost none left for $m = 7$

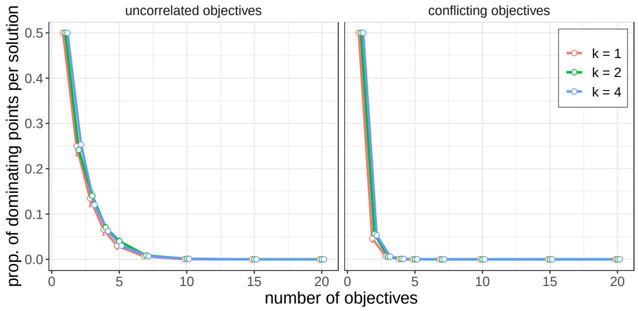

Figure 4: Average number of solutions dominating each solution.

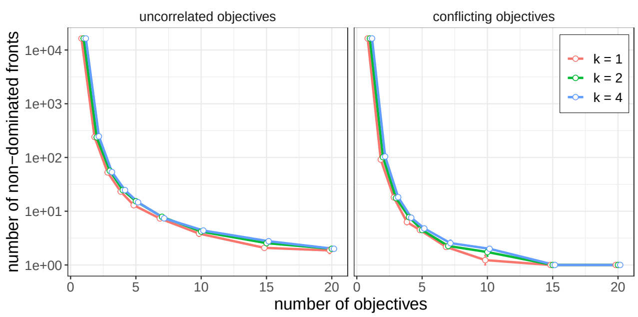

Figure 5: Number of non-dominated fronts.

uncorrelated objectives (or $m = 3$ conflicting objectives). This result suggests that most of non-optimal solutions are dominated by a single point in these cases, that is, almost every randomly sampled solution is either Pareto-optimal or dominated by just one solution.

Next, we follow another ranking strategy based on non-dominated sorting [11], as used, e.g., in NSGA-II [4]. All solutions are organized into different layers of mutually non-dominated solutions, and the rank of a solution corresponds to the layer it belongs to, such that a lower rank is better and Pareto-optimal solutions have a rank of 1. The number of non-dominated fronts (layers), or equivalently the maximum rank, is given in Figure 5 (notice the log-scale on the y-axis). As above, this significantly decreases with the number of objectives for uncorrelated objectives, and even more so for conflicting objectives. For 1 objective, there is one layer (rank) per solution. But for many objectives, all solutions have a rank very close to that of Pareto-optimal solutions, and most solutions belong to the first front (i.e., the Pareto front). Overall, as m increases, so does the probability of being non-dominated, and the expected rank of solutions is better.

This analysis not only confirms previous results on the ineffectiveness of Pareto dominance for solving many-objective problems [14], it also highlights that this ineffectiveness actually means that finding Pareto-optimal solutions becomes easier (or even trivial) with increasing number of objectives. It is particularly striking how little influence the setting of k has on this conclusion. Verel

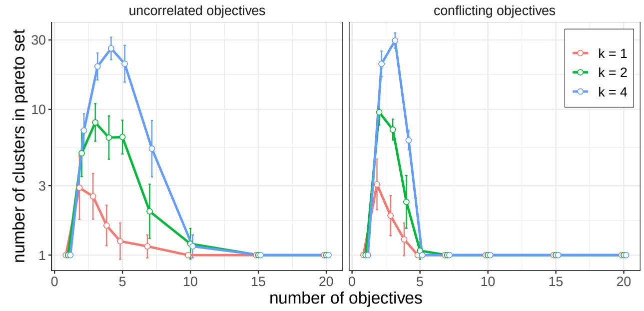

Figure 6: Number of connected components in the Pareto graph.

et al. [36] showed that there are slightly fewer Pareto-optimal solutions when k is large. Here we see this is true up to a certain number of objectives, after which all solutions are Pareto-optimal.

Finally, we argue that, if almost all solutions from the search space are Pareto-optimal, the challenge is no longer to optimize the objectives but to find a good representation of the Pareto front of tractable size, i.e., bounded archiving [19].

4 CONNECTEDNESS OF PARETO-OPTIMAL SOLUTIONS

The previous results deal with the dominance relation among solutions based on their position in the objective space, independently of their proximity in the solution space. Let us now take a closer look at the topology of the Pareto set based on the *connectedness* among Pareto-optimal solutions [7, 12]. We define the *Pareto graph* such that each node is a Pareto-optimal solution, and an edge connects two nodes if the corresponding solutions are *neighbors* in the solution space. For the considered ρmnk-landscapes, we say that two solutions are neighbors if they are separated by a Hamming distance of 1, which directly relates to the 1-bit-flip operator. If the Pareto graph is connected, i.e. there is a path between any pair of nodes, the Pareto set is connected [7, 12], which makes it possible for a local search to identify the whole Pareto set by starting with one Pareto-optimal solution. As shown in [24, 31], the degree of connectedness impacts the performance of multi-objective algorithms such as Pareto local search [29]. Intuitively, a large number of small connected components makes the optimization more difficult as an algorithm would have to jump from one component to another in order to make progress.

Figure 6 gives the number of connected components (CC) in the Pareto graph [31]. In the case where the value is 1, all Pareto-optimal solutions belong to the same CC, and therefore the Pareto set is connected. Conversely, when there are as many CC as Pareto-optimal solutions, all the nodes are isolated in the Pareto graph. For 1 objective, there is a single optimum so it naturally belongs to a single CC. We observe that the number of CC starts by increasing for a relatively small number of objectives, and then suddenly drops down to 1 for 15+ uncorrelated objectives, or 7+ conflicting objectives. In fact, there are already less than 2 CC on average for 10 uncorrelated objectives and 5 conflicting objectives, which makes it even more striking. Before this sudden drop, the problem

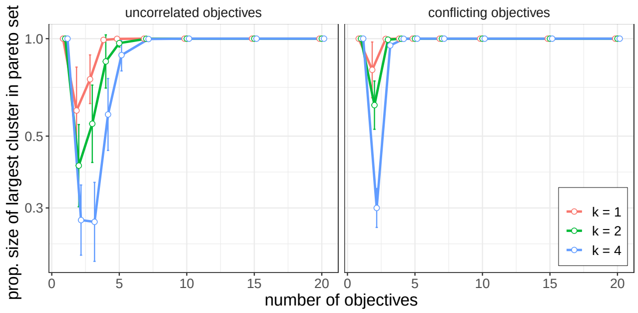

Figure 7: Proportional size of largest connected component.

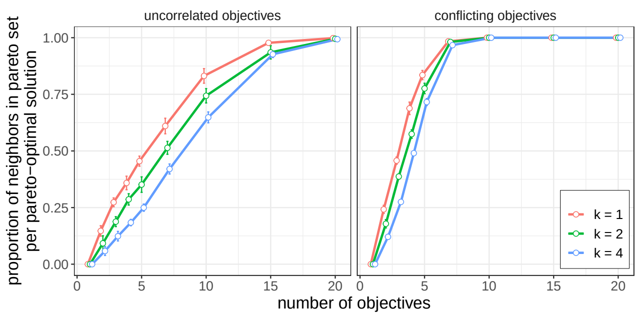

Figure 8: Average degree of nodes in the Pareto graph.

non-linearity noticeably affects the number of CC. In particular, larger k−values seem to break the connectedness, which is one of the reasons why larger k-values make the problems harder, which is a known result from 2 to 5 objectives [36]. However, after the drop, k has no effect and cannot break the connectedness. The above results are confirmed by Figure 7 where the proportional size of the largest CC [24, 36] is shown. Here we further see that for 7+ uncorrelated objectives or 4+ conflicting objectives, the largest CC contains almost all Pareto-optimal solutions. We can therefore deduce that the other CC contain only a handful of solutions. We finally report the average node degree of the Pareto graph in Figure 8; that is, for each Pareto-optimal solution, how many of its neighbors are other Pareto-optimal solutions. We normalize the values by the neighborhood size. The number of objectives seems to have a positive effect on the number of neighbors that belong to the Pareto graph. In fact, all neighbors of all nodes map to Pareto-optimal solutions for 20 uncorrelated objectives or 7+ conflicting objectives, that is, all neighbors of Pareto-optimal solutions are themselves Pareto-optimal.

Altogether, these results suggest that for many objectives, there are fewer connected components in the Pareto set, that Pareto-optimal solutions tend to belong to the same connected component, and that most neighbors from Pareto-optimal solutions are also Pareto-optimal. From this section and the previous one, we can conclude that Pareto-optimal solutions can be found more easily, and that it becomes more likely to find more Pareto-optimal solutions from other Pareto-optimal solutions when there are more objectives.

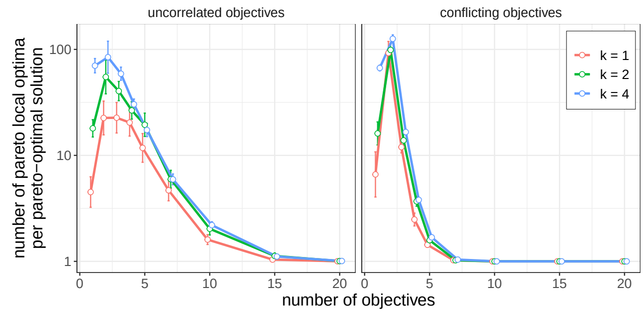

Figure 9: Ratio of Pareto local optima to Pareto-optimal solutions.

5 MULTIMODALITY

We now move on to the last part of our analysis, in which we study the multimodality of many-objective problems. By multimodality, we mean the number of Pareto local optimal solutions and of Pareto local optimal sets as found by multi-objective local search algorithms.

5.1 Local Optimal Solutions

We have previously commented on the structure induced by the neighborhood within the set of Pareto-optimal solutions. Let us now consider the landscape induced by the neighborhood within the whole solution space. Similar to single-objective optimization, a solution is a Pareto local optimal solution if it is not dominated by any of its neighbors [29]. As above, we consider the 1-bit-flip neighborhood for ρmnk-landscapes. Global optima (i.e. Pareto-optimal solutions) are also local optima (i.e. Pareto local optimal solutions), but not necessarily the other way around.

We report the number of local optima per global optimum, i.e., the number of Pareto local optimal solutions divided by the number of Pareto-optimal solutions, in Figure 9 for the different considered problems. If this ratio is 1, this means that all local optima are in fact global optima, in which case the multimodality is low. By contrast, the larger this ratio, the more (non-global) local optima and the higher the multimodality, which we expect to make the problem harder for local search algorithms. As expected, there are more local optima when k is large. This is consistent with known results for single-objective nk-landscapes [18, 37] and for ρmnk-landscapes with 2 to 5 objectives [36]. That said, here again the number of objectives seems to play a more important role. Indeed, we observe that the ratio increases from 1 to 2 objectives, and then significantly decreases (notice the log-scale) as there are more objectives. For uncorrelated objectives, most local optima are global optima for $m = 15$, and all of them are for $m = 20$. For conflicting objectives, this is the case already for $m = 7$ objectives.

A different view of the same observation is given in Figure 10, which reports the proportion of Pareto-optimal solutions (global optima), of Pareto local optima that are not Pareto-optimal (local optima) and of other (neither global nor local optimal) solutions. The green line in the plot shows how the proportion of local optima that are not global optima increases with the number of objectives, reaching a maximum at $m = 6$ (for uncorrelated objectives) or

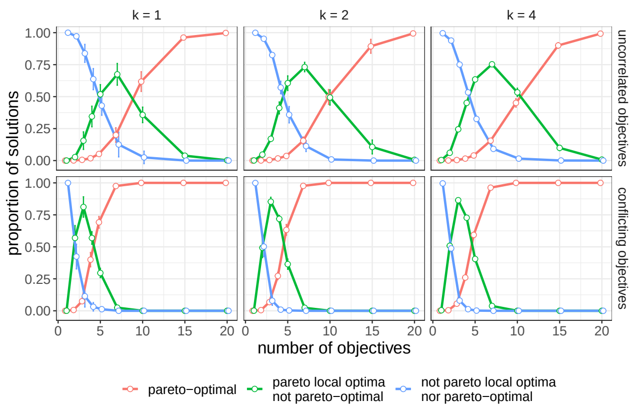

Figure 10: Proportion of Pareto-optimal solutions, of Pareto local optima (not Pareto), and of other solutions.

$m = 3$ (for conflicting objectives). However, for higher number of objectives, this value decreases until reaching zero, which means that all local optima are in fact global optima.

In summary, the more the objectives, the fewer the number of Pareto local optimal solutions per Pareto-optimal solution, so that the multimodality decreases. In fact, for many objectives, most (if not all) local optima are global optima. This knowledge can simplify the search. For example, an optimization algorithm could avoid checking the Pareto-optimality of Pareto local optimal solutions, since any local optima is unlikely to be dominated by any other solution.

5.2 Local Search and Local Optimal Sets

We conclude our analysis with a study related to the basins of attraction of multi-objective local search algorithms. A typical example of such algorithms is Pareto local search (PLS) [29]. PLS maintains an unbounded archive of mutually non-dominated solutions, which is initialized with a random solution. At each step, one solution is randomly selected from the archive, and all its neighbors are evaluated and compared against the archive. The current solution is thus tagged as *visited*. Non-dominated neighbors are added to the archive, and dominated solutions from the archive are filtered. PLS stops once all solutions from the archive are tagged as *visited*. A similar example of multi-objective local search is SEMO (simple evolutionary multi-objective optimizer) [21]. The only difference with PLS is that only one neighbor is evaluated at each step, randomly generated with repetition, such that SEMO does not detect when no more improvement is possible. Paquete et al. [29] define the attraction set of local search algorithms such as PLS and SEMO as follows: A *Pareto local optimal set* contains Pareto local optimal solutions only, such that all solutions are mutually non-dominated, and any neighbor of any solution is weakly dominated by a solution in the set [29]. Upon completion, both PLS and SEMO return such a Pareto local optimal set.

We investigate the performance of PLS and SEMO for $m \in \{2, 3, 4, 5\}$ objectives. For comparison, we also consider a simple random search, and a simple enumerative search where solutions are generated following a random order, but without repetition. We report the proportional number of identified Pareto-optimal solutions and the hypervolume [39] relative deviation with respect to the number of evaluated solutions (denoted as iterations). The algorithms were implemented in C++ using `Paradiseo` [5], and the hypervolume computation was performed using the implementation from [9] as provided within the `mco` R package [27]. We could not run the algorithms for larger values of m because the archiving and hypervolume tasks are too CPU-intensive. We stress that these sub-procedures are not directly related to the search process and do not impact the number of evaluations they perform. That said, the results reported below are sufficient to support our observations and findings for the considered problem size.

The results of all algorithms for uncorrelated objectives are reported in Figure 11. Notice the log-scale on the x-axis (iterations) for both the proportion of identified Pareto-optimal solutions (left) and the hypervolume relative deviation (right), and on the y-axis for hypervolume. As expected, the convergence profile of PLS and SEMO is very similar. Although SEMO is slightly outperformed at some intermediate stages of the search, it eventually catches up to PLS and finishes with the same level of approximation quality (except for $k = 4$ and $m = 5$). Correspondingly, random search and enumerative search are almost indistinguishable until about 5 000 iterations, and then the latter takes over by taking advantage of not evaluating the same solution twice. Since the maximum number of iterations is set as the solution space size ($2^n = 2^{14} = 16\,384$ evaluations), enumerative search always results in the Pareto set. We observe that PLS and SEMO are able to reach the Pareto set for $k = 1$ and $m \geq 4$ objectives, and are very close to it for $k = 2$ and $m = 5$ objectives. This suggests that there is only one Pareto local optimal set in this case: the Pareto set.

Looking in more detail at the results of PLS (left) and SEMO (right) in Figs. 12–13, this time for both uncorrelated and conflicting objectives, we see more clearly the impact of the number of objectives. In particular, for a given problem non-linearity, both algorithms are more quickly stuck into a lower-quality Pareto local optimal set when there are fewer objectives. This suggests that there are fewer Pareto local optimal sets for many-objective problems. Of course, the problem non-linearity also increases the landscape multimodality, but for a given $k-$value this decreases with m.

At last, we report in Figure 14 the number of iterations performed by PLS until it does not improve anymore (up to 2^n iterations), this time for 2 to 10 objectives. For uncorrelated objectives, PLS evaluates less than 5% of the solution space before getting stuck for 2 objectives, about 25% for 4 objectives, 50–75% for 5 objectives, and 100% for 7+ objectives. For conflicting objectives, this goes from 10% for 2 objectives, to 90% for 3 objectives, and 100% for 4+ objectives. These trends are almost independent of the problem non-linearity (k). Once again, this observation corroborates that the number of Pareto local optimal sets decreases with the number of objectives, such that multi-objective landscapes appear to be *more* multimodal than many-objective landscapes.

6 CONCLUSIONS

In this paper, we questioned the assumption that multi-objective combinatorial optimization problems (with 2 or 3 objectives) are easier to optimize than many-objective ones (with 4+ objectives).

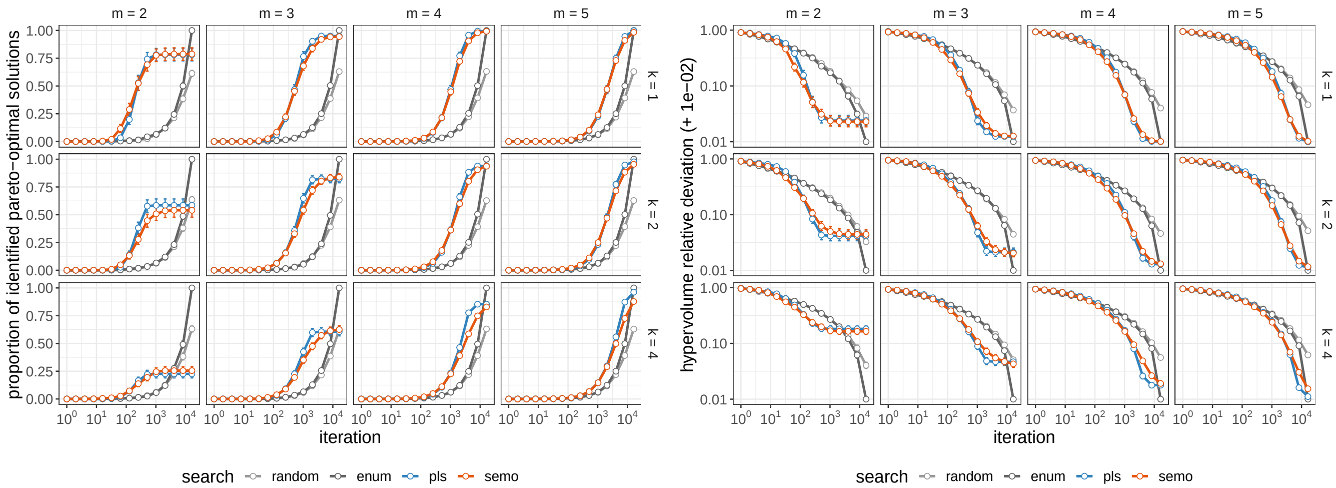

Figure 11: Proportion of identified Pareto-optimal solutions (left) and hypervolume relative deviation (right) for uncorrelated objectives.

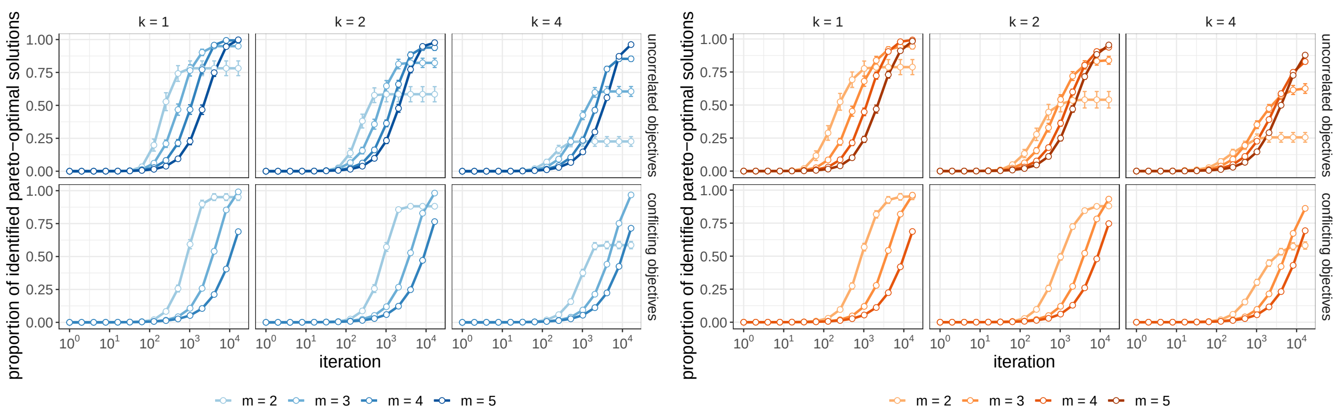

Figure 12: Proportion of Pareto-optimal solutions identified by PLS (left) and SEMO (right).

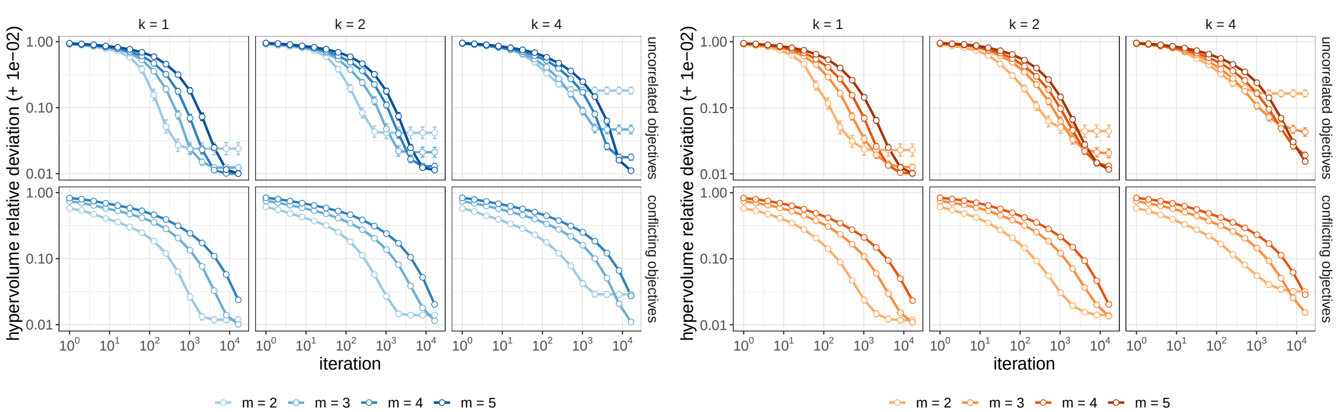

Figure 13: Hypervolume relative deviation for PLS (left) and SEMO (right).

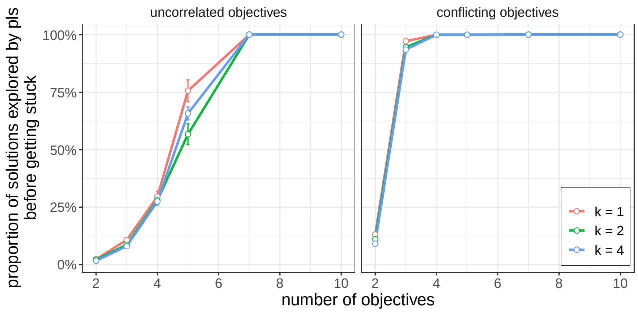

Figure 14: Number of solutions explored by PLS before reaching its natural stopping condition.

Experiments were conducted on a broad range of small-size ρmnk-landscapes with 2 to 20 objectives, and varying degrees of ruggedness and of conflict between the objectives. Our main findings can be summarized as follows. When there are *more* objectives:

(1) The conflict between (anti-correlated) objectives decreases, i.e., the more the objectives, the less conflicting they can be.
(2) The probability for a randomly-sampled solution to be Pareto-optimal increases, which implies that finding at least one Pareto-optimal solution becomes easier.
(3) The number of dominating points per solution decreases. As such, the small fraction of non-optimal solutions are in fact dominated by very few solutions.
(4) The number of non-dominated fronts decreases, which means that if we find a dominated solution, any solution that dominates it is likely to be Pareto-optimal.
(5) The connectedness of the Pareto set increases, in which case finding more Pareto-optimal solutions with local search becomes trivial, and finding the complete Pareto set is limited only by the intractability of its size, and not by any difficultly induced by its landscape.
(6) The number of local optima per Pareto-optimal solution decreases, which means that any solution identified as a local optima is likely to be Pareto-optimal.
(7) The proportion of Pareto-optimal solutions identified by local search increases, i.e., with more objectives, a local search identifies a larger fraction of the Pareto front.
(8) The relative hypervolume covered by local search increases. If run until completion, local search will therefore be more effective for many-objective problems than multi-objective ones.
(9) The number of solutions explored by local search before converging to a local optimal sets increases, which means that the search does not easily get stuck into local optima and no mechanism is needed for escaping them.
(10) The number of local optimal sets, which is a measure of multimodality, decreases. High multimodality is known to make a problem harder to solve [10, 26]. In other words, our empirical results reveal that multi-objective problems are in fact *more multimodal* than many-objective problems.

The above observations suggest that, after 4 or 5 objectives, adding more objectives to an optimization problem typically makes the goal of finding a Pareto set approximation easier, almost independently from the ruggedness of the landscape. Unlike the single- and multi-objective case, many-objective local search is rarely trapped into local optima and converges to better approximations. From the optimization point of view, the challenge of many-objective approaches is not to accelerate the convergence towards the Pareto front, nor to identify which solutions are Pareto-optimal, and neither to escape from attraction points. It is rather to avoid revisiting solutions or regions of the Pareto front already explored, and to identify a subset of the Pareto front that maximizes some quality metric or the decision-maker's preferences.

Nevertheless, beyond the number of solutions explored, many-objective optimization presents other challenges, such as the computational effort to handle additional objectives or the difficulty of processing and maintaining numerous solutions mapping to high-dimensional objective spaces. Many-objective algorithms would obviously benefit from any progress on CPU-intensive procedures such as solution ranking, online archiving or the computation of set quality indicators. Such progress would additionally allow us to validate our results on broader benchmarks, and to experiment with bounded-size local optimal sets (i.e., approximations) [25] and representations of the Pareto set [35] relative to the best subset of a given size. Furthermore, we plan to investigate different problem classes, including continuous multi- and many-objective functions, and to consider additional landscape features such as the solution space dimension or other types of dominance relations. Lastly, many-objective optimizers often consider scalarization functions, and it remains an open question whether the landscape of such functions becomes harder or easier to optimize when adding more objectives.

Reproducibility. Relevant data is available at the following URL: https://doi.org/10.5281/zenodo.7625398.

REFERENCES

[1] Hernán E. 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. https://doi.org/10.1016/j.ejor.2006.08.004
[2] Dimo Brockhoff, Tobias Friedrich, N. Hebbinghaus, C. Klein, Frank Neumann, and Eckart Zitzler. 2007. Do Additional Objectives Make a Problem Harder? In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2007*, Dirk Thierens et al. (Eds.). ACM Press, New York, NY, 765–772. https://doi.org/10.1145/1276958.1277114
[3] Fabio Daolio, Arnaud Liefooghe, Sébastien Verel, Hernán E. Aguirre, and Kiyoshi Tanaka. 2017. Problem Features versus Algorithm Performance on Rugged Multiobjective Combinatorial Fitness Landscapes. *Evolutionary Computation* 25, 4 (2017), 555–585. https://doi.org/10.1162/evco_a_00193
[4] Kalyanmoy Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197. https://doi.org/10.1109/4235.996017
[5] Johann Dréo, Arnaud Liefooghe, Sébastien Verel, Marc Schoenauer, Juan-Julián Merelo, Alexandre Quemy, Benjamin Bouvier, and Jan Gmys. 2021. Paradiseo: from a modular framework for evolutionary computation to the automated design of metaheuristics. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO Companion 2021*, Francisco Chicano and Krzysztof Krawiec (Eds.). ACM Press, New York, NY. https://doi.org/10.1145/3449726.3463276
[6] Matthias Ehrgott. 2005. *Multicriteria Optimization* (2nd ed.). Springer, Berlin, Germany. https://doi.org/10.1007/3-540-27659-9
[7] Matthias Ehrgott and Kathrin Klamroth. 1997. Connectedness of Efficient Solutions in Multiple Criteria Combinatorial Optimization. *European Journal of Operational Research* 97, 1 (1997), 159–166. https://doi.org/10.1016/S0377-2217(96)00116-6
[8] Carlos M. Fonseca and Peter J. Fleming. 1993. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In *Proceedings*

of the Fifth International Conference on Genetic Algorithms (ICGA'93), Stephanie Forrest (Ed.). Morgan Kaufmann Publishers, 416–423.

[9] Carlos M. Fonseca, Luís Paquete, and Manuel López-Ibáñez. 2006. An improved dimension-sweep algorithm for the hypervolume indicator. In *Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006)*. IEEE Press, Piscataway, NJ, 1157–1163. https://doi.org/10.1109/CEC.2006.1688440

[10] Josselin Garnier and Leila Kallel. 2001. Efficiency of Local Search with Multiple Local Optima. *SIAM Journal Discrete Mathematics* 15, 1 (2001), 122–141. https://doi.org/10.1137/S0895480199355225

[11] David E. Goldberg. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Boston, MA.

[12] Jochen Gorski, Kathrin Klamroth, and Stefan Ruzika. 2011. Connectedness of Efficient Solutions in Multiple Objective Combinatorial Optimization. *Journal of Optimization Theory and Applications* 150, 3 (2011), 475–497. https://doi.org/10.1007/s10957-011-9849-8

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

[14] Kokolo Ikeda, Hajime Kita, and Shigenobu Kobayashi. 2001. Failure of Pareto-based MOEAs: Does non-dominated really mean near to optimal? In *Proceedings of the 2001 Congress on Evolutionary Computation (CEC'01)*. IEEE Press, Piscataway, NJ, 957–962.

[15] Hisao Ishibuchi, N. Tsukamoto, and Y. Nojima. 2008. Evolutionary many-objective optimization: A short review. In *Proceedings of the 2008 Congress on Evolutionary Computation (CEC 2008)*. IEEE Press, Piscataway, NJ, 2419–2426. https://doi.org/10.1109/CEC.2008.4631121

[16] M. T. Jensen. 2003. Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms. *IEEE Transactions on Evolutionary Computation* 7, 5 (2003), 503–515.

[17] M. T. Jensen. 2004. Helper-Objectives: Using Multi-Objective Evolutionary Algorithms for Single-Objective Optimisation. *Journal of Mathematical Modelling and Algorithms* 3, 4 (2004), 323–347.

[18] S. A. Kauffman. 1993. *The Origins of Order*. Oxford University Press.

[19] Joshua D. Knowles and David Corne. 2004. Bounded Pareto Archiving: Theory and Practice. In *Metaheuristics for Multiobjective Optimisation*, Xavier Gandibleux, Marc Sevaux, Kenneth Sörensen, and V. T'Kindt (Eds.). Lecture Notes in Economics and Mathematical Systems, Vol. 535. Springer, Berlin, Germany, 39–64. https://doi.org/10.1007/978-3-642-17144-4_2

[20] Joshua D. Knowles, Richard A. Watson, and David Corne. 2001. Reducing Local Optima in Single-Objective Problems by Multi-objectivization. In *Evolutionary Multi-criterion Optimization, EMO 2001*, Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne (Eds.). Lecture Notes in Computer Science, Vol. 1993. Springer, Heidelberg, 269–283. https://doi.org/10.1007/3-540-44719-9_19

[21] Marco Laumanns, Lothar Thiele, and Eckart Zitzler. 2004. Running time analysis of evolutionary algorithms on a simplified multiobjective knapsack problem. *Natural Computing* 3, 1 (2004), 37–51.

[22] Bingdong Li, Jinlong Li, Ke Tang, and Xin Yao. 2015. Many-Objective Evolutionary Algorithms: A Survey. *Comput. Surveys* 48, 1 (2015), 1–35. https://doi.org/10.1145/2792984

[23] Arnaud Liefooghe, Fabio Daolio, Bilel Derbel, Sébastien Verel, Hernán E. Aguirre, and Kiyoshi Tanaka. 2020. Landscape-Aware Performance Prediction for Evolutionary Multi-objective Optimization. *IEEE Transactions on Evolutionary Computation* 24, 6 (2020), 1063–1077.

[24] Arnaud Liefooghe, Bilel Derbel, Sébastien Verel, Hernán E. Aguirre, and Kiyoshi Tanaka. 2013. What Makes an Instance Difficult for Black-box 0–1 Evolutionary Multiobjective Optimizers? In *Artificial Evolution: 11th International Conference, Evolution Artificielle, EA, 2013*, Pierrick Legrand et al. (Eds.). Lecture Notes in Computer Science, Vol. 8752. Springer, Heidelberg, 3–15. https://doi.org/10.1007/978-3-319-11683-9_1

[25] Arnaud Liefooghe, Manuel López-Ibáñez, Luís Paquete, and Sébastien Verel. 2018. Dominance, Epsilon, and Hypervolume Local Optimal Sets in Multi-objective Optimization, and How to Tell the Difference. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2018*, Hernán E. Aguirre and Keiki Takadama (Eds.). ACM Press, New York, NY, 324–331. https://doi.org/10.1145/3205455.3205572

[26] Katherine M. Malan and Andries Engelbrecht. 2013. A survey of techniques for characterising fitness landscapes and some possible ways forward. *Information Sciences* 241 (2013), 148–163. https://doi.org/10.1016/j.ins.2013.04.015

[27] Olaf Mersmann. 2014. *mco: Multiple Criteria Optimization Algorithms and Related Functions*. http://CRAN.R-project.org/package=mco R package version 1.0-15.1.

[28] John Mote, Ishwar Murthy, and David L. Olson. 1991. A parametric approach to solving bicriterion shortest path problems. *European Journal of Operational Research* 53, 1 (1991), 81–92. https://doi.org/10.1016/0377-2217(91)90094-C

[29] Luís Paquete, Tommaso Schiavinotto, and Thomas Stützle. 2007. On Local Optima in Multiobjective Combinatorial Optimization Problems. *Annals of Operations Research* 156 (2007), 83–97. https://doi.org/10.1007/s10479-007-0230-0

[30] Luís Paquete and Thomas Stützle. 2006. A study of stochastic local search algorithms for the biobjective QAP with correlated flow matrices. *European Journal of Operational Research* 169, 3 (2006), 943–959.

[31] Luís Paquete and Thomas Stützle. 2009. Clusters of non-dominated solutions in multiobjective combinatorial optimization: An experimental analysis. In *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, V. Barichard, M. Ehrgott, Xavier Gandibleux, and V. T'Kindt (Eds.). Lecture Notes in Economics and Mathematical Systems, Vol. 618. Springer, Berlin, Germany, 69–77. https://doi.org/10.1007/978-3-540-85646-7

[32] Luís Paquete, Thomas Stützle, and Manuel López-Ibáñez. 2007. Using experimental design to analyze stochastic local search algorithms for multiobjective problems. In *Metaheuristics: Progress in Complex Systems Optimization*, Karl F. Doerner, Michel Gendreau, Peter Greistorfer, Walter J. Gutjahr, Richard F. Hartl, and Marc Reimann (Eds.). Operations Research / Computer Science Interfaces, Vol. 39. Springer, New York, NY, 325–344. https://doi.org/10.1007/978-0-387-71921-4_17

[33] Robin C. Purshouse and Peter J. Fleming. 2007. On the Evolutionary Optimization of Many Conflicting Objectives. *IEEE Transactions on Evolutionary Computation* 11, 6 (2007), 770–784. https://doi.org/10.1109/TEVC.2007.910138

[34] Riccardo Rebonato and Peter Jäckel. 1999. The most general methodology to create a valid correlation matrix for risk management and option pricing purposes. *Journal of Risk* 2 (1999), 17–28.

[35] Daniel Vaz, Luís Paquete, Carlos M. Fonseca, Kathrin Klamroth, and Michael Stiglmayr. 2015. Representation of the non-dominated set in biobjective discrete optimization. *Computers & Operations Research* 63 (2015), 172–186. https://doi.org/10.1016/j.cor.2015.05.003

[36] 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. https://doi.org/10.1016/j.ejor.2012.12.019

[37] Edward D. Weinberger. 1991. Local properties of Kauffman's N-k model: A tunably rugged energy landscape. *Physical Review A* 44, 10 (1991), 6399.

[38] P. Winkler. 1985. Random Orders. *Order* 1 (1985), 317–331.

[39] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M. Fonseca, and Viviane Grunert da Fonseca. 2003. Performance Assessment of Multiobjective Optimizers: an Analysis and Review. *IEEE Transactions on Evolutionary Computation* 7, 2 (2003), 117–132. https://doi.org/10.1109/TEVC.2003.810758