

# Search Trajectories Networks of Multiobjective Evolutionary Algorithms

Yuri Lavinás<sup>1</sup> , Claus Aranha<sup>2</sup>, and Gabriela Ochoa<sup>3</sup> 

<sup>1</sup> University of Tsukuba, Tsukuba, Japan,  
lavinás.yuri.xp@alumni.tsukuba.ac.jp

<sup>2</sup> University of Tsukuba, Tsukuba, Japan,  
caranha@cs.tsukuba.ac.jp

<sup>3</sup> University of Stirling, Stirling, UK  
gabriela.ochoa@cs.stir.ac.uk

**Abstract.** Understanding the search dynamics of multiobjective evolutionary algorithms (MOEAs) is still an open problem. This paper extends a recent network-based tool, search trajectory networks (STNs), to model the behavior of MOEAs. Our approach uses the idea of decomposition, where a multiobjective problem is transformed into several single-objective problems. We show that STNs can be used to model and distinguish the search behavior of two popular multiobjective algorithms, MOEA/D and NSGA-II, using 10 continuous benchmark problems with 2 and 3 objectives. Our findings suggest that we can improve our understanding of MOEAs using STNs for algorithm analysis.

**Keywords:** algorithm analysis, search trajectories, continuous optimization, visualization, multi-objective optimization

## 1 Introduction

Most real-world optimization problems involve multiple conflicting objectives. This has prompted the development of a variety of multiobjective evolutionary algorithms (MOEAs), which can be classified into three broad categories, based on dominance [5], indicators [1] and decomposition [21]. There has been significant progress in improving MOEAs in all categories, and these algorithms are widely used in practice. However, algorithm development and improvement are mostly guided by intuition and empirical performance comparisons. We argue that there is a lack of accessible tools to analyze, contrast and visualize the dynamic behavior of MOEAs.

The main goal of this article is to generalize to multiobjective optimization a recent graph-based modeling tool, search trajectory networks (STNs), which was originally proposed for single objective optimization [14, 15]. Our approach uses decomposition, a key strategy in multiobjective optimization, where the multiobjective problem is transformed into several single-objective problems. This is convenient as it allows us to use the existing tools and features proposed for single objective STNs [14, 15].

This study is exploratory and interpretative in nature, taking the form of a case study using 10 continuous benchmark problems with 2 and 3 objectives and two broadly known MOEAs, MOEA/D [21] and NSGA-II [5]. To the best of our knowledge, this is the first effort to apply STNs for modeling the search behavior of MOEAs.

The paper is organized as follows. Section 2 overviews previous work related to visualization in multiobjective optimization. Section 3 introduces relevant concepts. Our proposal to extend STNs to model MOEAs is described in Section 4. The experimental setup and results are presented in Sections 5 and 6, respectively. Finally, Section 7 outlines our main findings and suggestions for future work.

## 2 Related Work

Most visualization approaches in the literature for multiobjective optimization focus entirely on the objective space, completely ignoring the decision space. The classical visualization shows the true or approximated Pareto front for 2 or 3 objectives in a standard scatter plot. Extensions to this idea for visualizing problems with more than three objectives, using dimensionality reduction techniques, have been proposed [19]. However, by purely focusing on the objective space, the interaction effects from the decision variables are ignored; therefore, almost no information on the structural properties of the problem landscapes can be derived.

Very few visualization techniques in the literature provide a joint view of the decision and objective spaces in continuous multiobjective optimization. Fonseca et al. [7] proposed the cost landscapes, which use dominance ranking to evaluate points in the decision space with respect to global optimal trade-offs. This approach, however, does not capture local optimal sets. Kerschke and Grimme [8] proposed the gradient field maps to explicitly address local optimal sets, with further extensions plotting landscapes with optimal trade-offs [16], and providing an accessible dashboard for visualization [17]. In combinatorial optimization, recent work has adapted the local optima networks model [13] to multiobjective optimization, providing visual insights into the distribution and connectivity pattern of Pareto local optimal solutions [11] and dominance-based hill-climbing [6]. These are insightful visual approaches; however, they concentrate on the structural configuration of local and global optima in the optimization landscapes, rather than on the dynamic (trajectory) behavior of the search process.

## 3 Preliminaries

### 3.1 Search Trajectory Networks

The original STN model definitions for single objective optimization can be found in [15]; we rephrase them here for completeness and to guide our proposed extension to multiobjective optimization. To define a network model, we need to specify their nodes and edges. The relevant definitions are given below.

**Representative solution.** Is a solution to the problem at a given iteration that represents the status of the search process. For population-based algorithms, the solution with the best fitness in the population at a given generation is chosen as the representative solution.

**Location.** Is a non-empty subset of solutions that results from a mapping process. Each solution in the search space is mapped to one location. Several similar solutions are generally mapped to the same location, as the locations represent a partition of the search space.

**Search trajectory.** Given a sequence of representative solutions in the order in which they are encountered during the search process, a search trajectory is defined as a sequence of locations formed by replacing each solution with its corresponding location.

**Node.** Is a location in a search trajectory of the search process being modeled. The set of nodes is denoted by  $N$ .

**Edges.** Edges are directed and connect two consecutive locations in the search trajectory. Edges are weighted with the number of times a transition between two given nodes occurred during the process of sampling and constructing the STN. The set of edges is denoted by  $E$ .

**Search trajectory network (STN).** Is a directed graph  $STN = (N, E)$ , with node set  $N$ , and edge set  $E$  as defined above.

The data to construct an STN model is gathered while the studied algorithm is running. Specifically, the required output from a run is a list of steps connecting two adjacent representative solutions in the search process. Each search step is stored as an entry in a log file containing the two consecutive representative solutions being linked with the step; these transitions become the edges of the network model. Once the data logs of a predefined number of runs of a given algorithm-problem pair are gathered, a post-processing maps solutions to locations, aggregates all the locations and transitions, and constructs a network object.

### 3.2 Multiobjective Optimisation Problems

In Multiobjective Optimization Problems (MOPs), a solution to the problem is evaluated by multiple objective functions, which are possibly conflicting. One key characteristic of MOPs is that a single solution rarely is the optimal solution for all objectives. This leads to the concept of “Pareto dominance”: A solution  $x_j$  is said to be dominated by  $x_i$  if  $x_i$  is better than  $x_j$  in one objective, and at least equivalent in all others. Otherwise,  $x_j$  is non-dominated by  $x_i$ . Note that two solutions can be mutually non-dominated (each solution is better in a different objective).

As a consequence of Pareto dominance, the search status of an MOEA cannot be represented by a single representative solution. Therefore, the definitions of location, node, edge, and search trajectory must be extended in a multiobjective context. Our proposed extension of STNs to multiobjective optimization relies on a few operations that are used in dominance-based MOEAs such as NSGA-II [5]

and decomposition-based MOEAs such as MOEA/D [21]. Brief descriptions of these operations are given below.

**Non-dominated sorting.** Sorts a set  $X$  of solutions based on their dominance relationship. Initially, we find the subset of solutions,  $X_{ND}$ , in  $X$  that are non-dominated in relation to all other solutions and give the solutions in  $X_{ND}$  rank 0. Then we repeat this procedure on the set  $X - X_{ND}$ , giving the new non-dominated solution set the rank 1. The procedure keeps repeating until we have ranked all solutions in  $X$ .

**Decomposition.** Breaks down a MOP into a set of single objective subproblems, where each subproblem is a combination of the objectives, characterized by a weight vector. There are several methods to generate a set of weight vectors (decomposition methods). In this work, we use the Uniform Design [20], as it allows us to choose the number of weight vectors to be generated explicitly.

**Scalar aggregation function.** Takes the objective values of one solution and the weight vector of one subproblem and calculates a scalar value that represents the quality of that solution for that particular subproblem. In a decomposition-based MOEA, these quality values are used to associate one representative solution to each subproblem. There are several scalar aggregation functions (also called scalarization functions), and in this work, we use the Weighted Tchebycheff [4, 12], which is less affected by the shape of the Pareto front in a given MOP.

## 4 STN Extension for the Multiobjective Domain

As discussed in the previous section, STNs are calculated based on the sequence of representative solutions in an algorithm execution. However, each iteration in an MOEA does not contain a single representative solution. In this section, we describe a method to calculate STNs for multiobjective optimization.

The key idea of this method is that we keep track of a small number of decomposition vectors, match a representative solution to a vector, and then merge the vector trajectories of each vector into a single multiobjective STN. The following processes describes the steps to associate a solution to each of the vectors:

1. Initially, we choose the number of decomposition vectors  $n$  and generate these vectors using the decomposition technique, using the technique discussed in the previous section.
2. Every iteration, we apply the non-dominated sorting procedure on the solution set and select the set of rank 0 solutions as the set of candidate solutions. If the number of solutions in this set is less than  $n$ , we add solutions from rank 1, 2, and so on until we have  $n$  or more candidate solutions.
3. Every iteration, we select a representative solution for each vector from the set of candidate solutions, using the scalar aggregation function. In the case of ties, we choose the newest solution, discarding solutions already associated with that vector.

After associating one representative solution for each weight vector, we map the locations of each representative solution as the nodes. To create this mapping, we use a precision parameter to portion the space into hypercubes with length  $10^{-03}$ . In other words, a node in the STN is equivalent to a  $0.001 \times 0.001 \times 0.001$  hypercube. Edges are given by the sequence of locations of consecutive iteration. Each decomposition vector is modeled as an STN. Finally, we merge these STNs, establishing the merged STN, that models the search progress of an algorithm in a problem. We merge the STNs as follows:

1. The merged STN model merges the  $n$  STNs of each decomposition vector and is obtained by the graph union of the  $n$  individual graphs.
2. The merged graph contains the nodes and edges present in at least one of the vectors graphs. Attributes are kept for the nodes and edges, indicating whether they were visited by both algorithms (shared) or by one of them only.

## 5 Experiments

We conduct the following experimental and exploratory study to investigate whether STNs can discriminate between MOEA/D-DE and NSGA-II. This analysis is conducted by characterizing and visualizing the search behavior of these MOEAs.

In this work, we use the UF benchmark set [23], with 2 and 3 objectives. Both algorithms were widely studied, and collectively these works have shown that MOEA/D performs better than NSGA-II in this group of problems. Thus the UF benchmark set helps show that the STN can discriminate these metaheuristics. For all functions in all benchmark sets, we set the number of dimensions to  $D = 10$ . The implementation of the test problems available from the *smoof* package [2] was used in all experiments. The UF Benchmark set comprises ten unconstrained test problems with Pareto sets designed to be challenging to existing algorithms [9]. Problems UF1-UF7 are two-objective MOPs, while UF8-UF10 are three-objective problems [23].

### 5.1 Experimental Parameters

We used the MOEA/D variant using the Differential Evolution (MOEA/D-DE) parameters as they were introduced in the work of Li and Zhang [10] in all tests. Details of these parameters can be found in the documentation of package MOEADr and the original MOEA/D-DE reference [3,4,22]. We used the NSGA-II parameters as they were introduced in the work of Deb et al. [5] in all tests, except for the use of the DE mutation operator, with the same parameters as MOEA/D. Details of these parameters can be found in the documentation of package nsga2R and the original reference [18]. Table (1) summarizes the experimental parameters for both algorithms.

Table 1: Experimental parameter settings.

MOEA/D Parameters	Value
DE mutation	$F = 0.25$
Polynomial mutation	$\eta_m = 20$ $prob = 0.01$
Restricted Update	$nr = 2$ $\delta_p = 0.9$
Neighborhood size	$T = 20\%$ of the pop. size
SLD weight vectors	250
NSGA-II Parameters	Value
Tournament Size	2
DE mutation	$F = 0.25$
Polynomial mutation	$\eta_m = 3$ $prob = 0.1$
Population size	Value
MOEA/D and NSGA-II	250
Experiment Parameters	Value
Repeated runs	3
Computational budget	30000 evals.

For creating the STNs, we use  $n = 5$  different vectors, and for each vector, there is one solution at a given iteration. Our idea here was to use a small number of vectors that could capture visually valuable information about the pair algorithm-problem. More work is required to understand the effect of different values for the number of vectors.

## 5.2 Metrics

*MOP metrics:* We use the following criteria to compare the results of the different strategies: (a) final approximation hypervolume (HV) and Inverted Generational Distance (IGD). For calculating the hypervolume, we set the reference point to (1.1,1.1) for two objective problems and (1.1,1.1,1.1) for three objective problems.

*STN metrics:* We use seven STN metrics to assess the global structure of the trajectories and bring insight into the behavior of the MOEAs modeled. These metrics are (1) the number of unique nodes and (2) the number of unique edges, and we also calculate (3) the number of shared nodes between vectors, (4) the number of Pareto optimal solutions, (5) the number of components of the network and (6) the mean and (7) maximum values of the in- and out-degree of

nodes. These metrics are summarised in Table 2. It is worth noting that additional metrics could also be considered, such as the distances between optimal solutions and centrality metrics for the optimal solutions.

Table 2: Description of STN metrics

Metric	Justification
Number of nodes	Shows unique locations visited.
Number of edges	Shows unique search transitions between nodes.
Number of shared nodes	Shows nodes visited by more than one components.
Number of optimal solutions	Shows what nodes are in the theoretical PF.
Number of Components	Show the distribution of the search progress.
Mean in/out-degree	Shows the mean degree of branching and loops.
Maximum in/out-degree	Shows the maximum degree of branching and loops.

### 5.3 Reproducibility

For reproducibility purposes, all the code and experimental scripts are available online at [https://github.com/yurilavinas/STN\\_MOP/tree/evostar](https://github.com/yurilavinas/STN_MOP/tree/evostar).

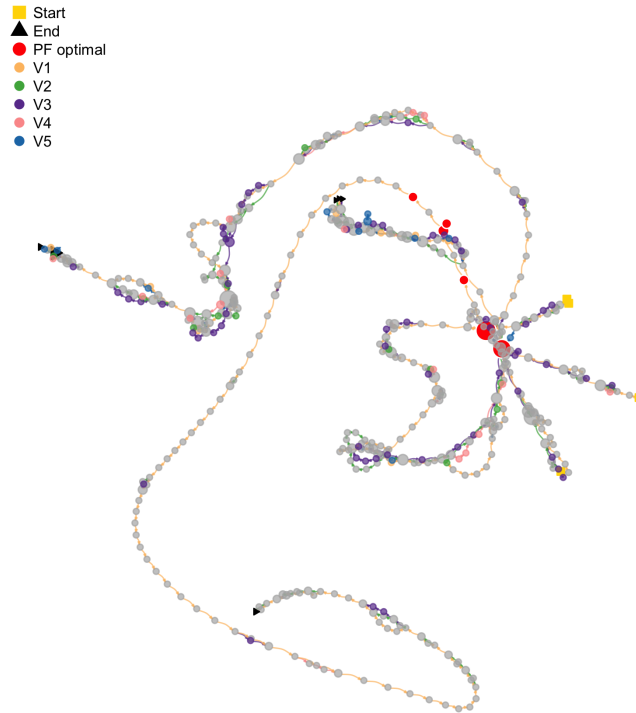
## 6 Results

This section compares the metaheuristic algorithms MOEA/D and NSGA-II in use of the STN model. For that, we compare the (a) STNs visualizations and (b) the performance in terms of MOP metrics, both HV and IGD, (c) and the STN metrics: (1) the number of nodes and (2) edges, (3) number of shared nodes, (4) the number of Pareto optimal solutions, (5) the number of components of the network, (6) the mean and (7) maximum in-degree and (8) the mean and (9) maximum out-degree.

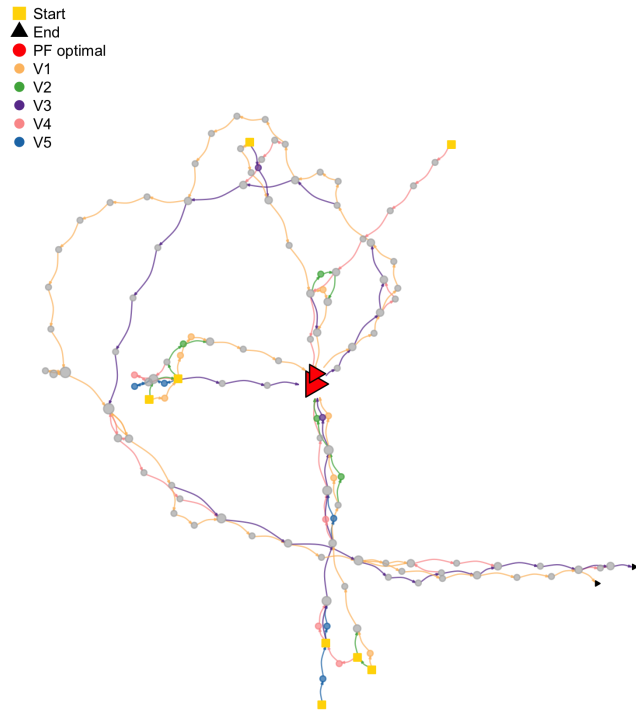
The visualisations in Figures 1, 2, 3 illustrate the STN by three independent runs of both candidate algorithms on the UF3, UF5 and UF8 benchmark MOPs. Figure 4 shows the number of nodes for each vector traversed by MOEA/D and NSGA-II on these same three MOPs. The first two MOPs are problems with two objectives, while UF8 has three objectives. Table 3 displays the traditional HV (higher is better) and IGD (lower is better) metrics and the STN metrics.

Considering the colors used in the STN visualizations, we use yellow squares to indicate the start of trajectories and black triangles to indicate the end of trajectories. The red color shows the best Pareto optimal solutions, and light grey circles show shared locations visited by more than one vector of the same algorithm. Finally, each vector has its color: light orange for V1; green for V2; purple for V3; light pink for V4; and light blue for V5.

In these visualizations, the size of nodes and the width of edges are proportional to how many times they were visited by the algorithms during the aggregation of runs used to extract the model. Moreover, these visualisations use



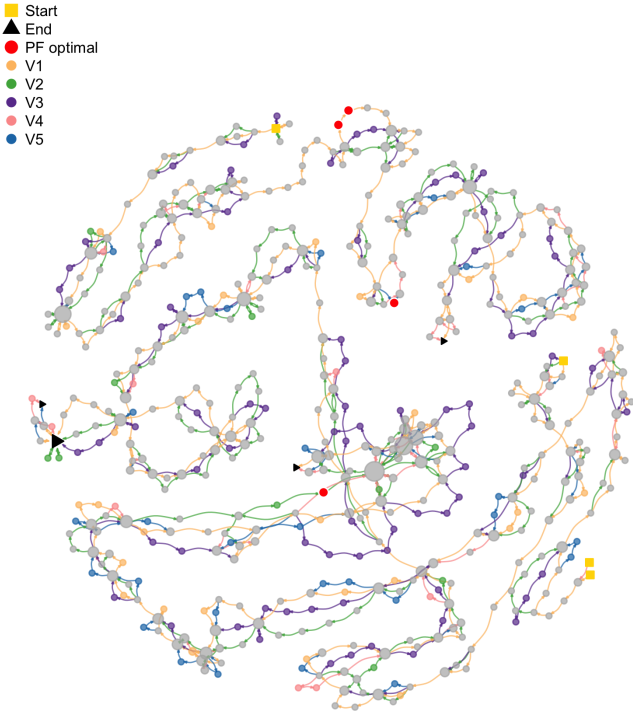
(a) STN of MOEA/D on UF3.



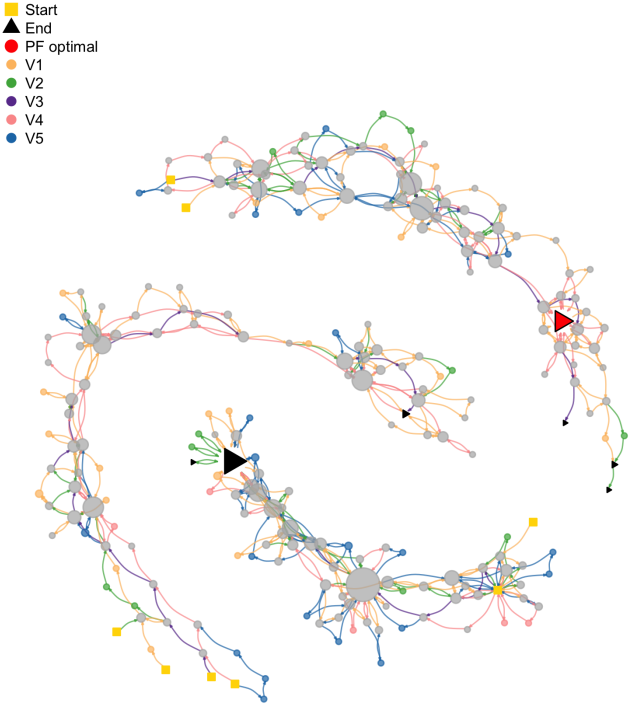
(b) STN of NSGA-II on UF3.

Fig. 1: We can see that MOEA/D (top) is able to find many optimal solutions while NSGA-II (bottom) finds optimal solutions at the end of their trajectory (red triangles). This reflects NSGA-II characteristic of using the non-dominance relationship as update criterion.



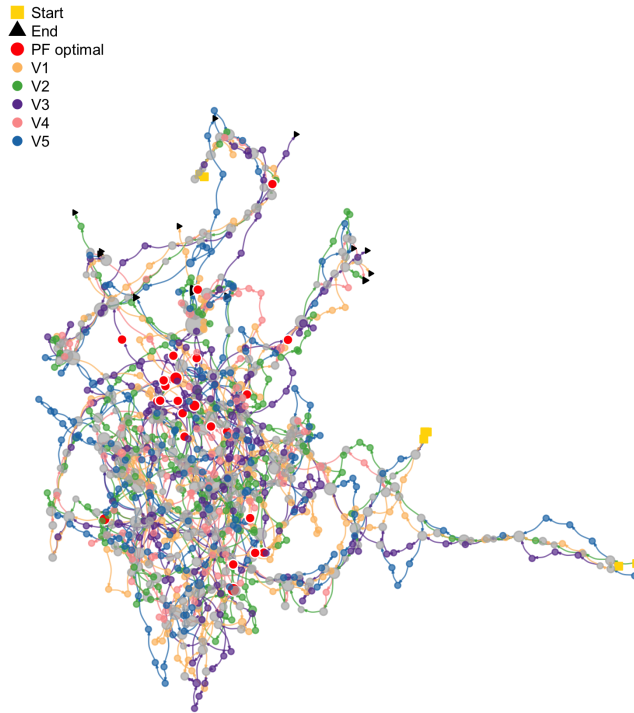


(a) STN of MOEA/D on UF5.

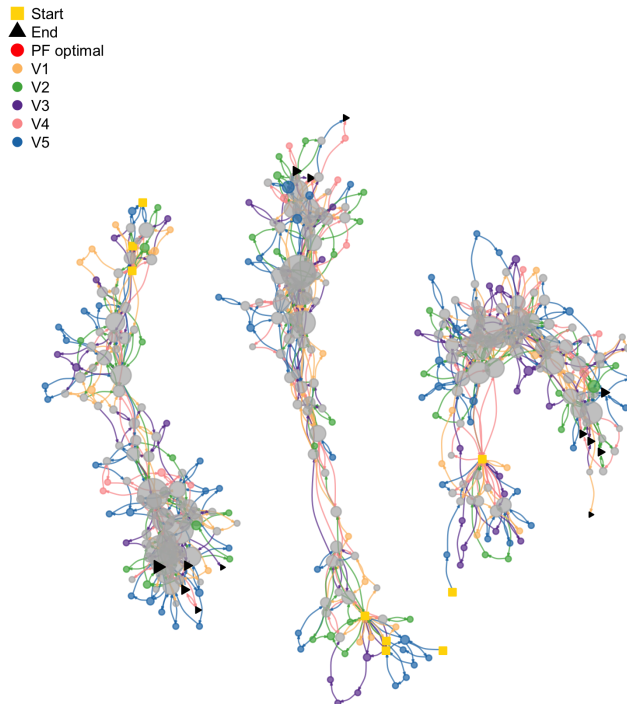


(b) STN of NSGA-II on UF5.

Fig. 2: MOEA/D (top) is able to find many optimal solutions, in two different decomposition vectors and NSGA-II (bottom) only finds optimal solutions in one of the three vectors.

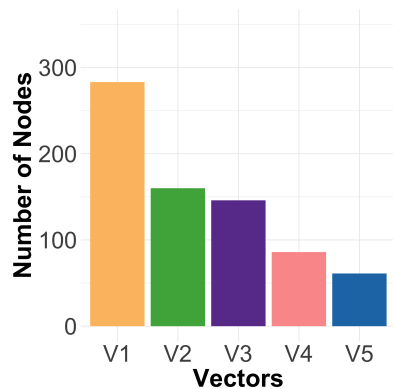


(a) STN of MOEA/D on UF8.

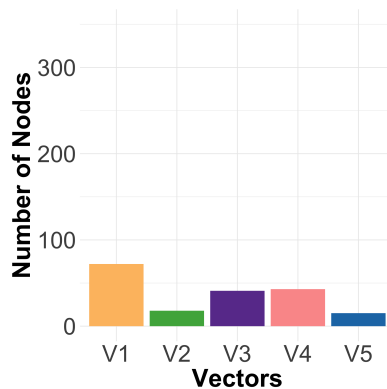


(b) STN of NSGA-II on UF8.

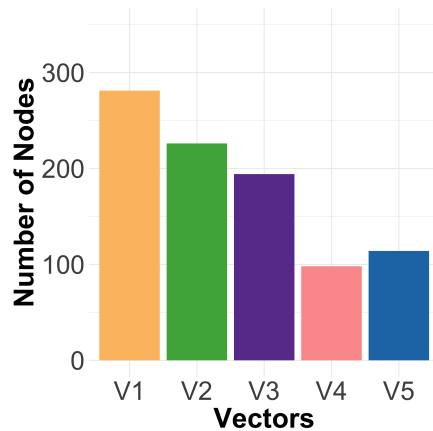
Fig. 3: MOEA/D finds many optimal solutions, around many different areas and NSGA-II (bottom) progresses the search with all vectors sharing nodes, from start to end.



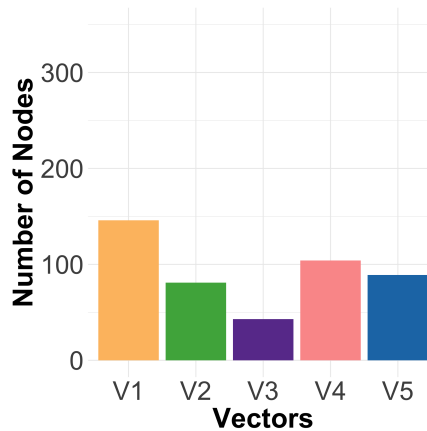
(a) MOEA/D on UF3.



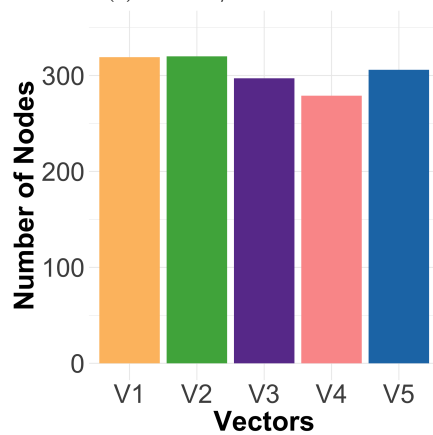
(b) NSGA-II on UF3.



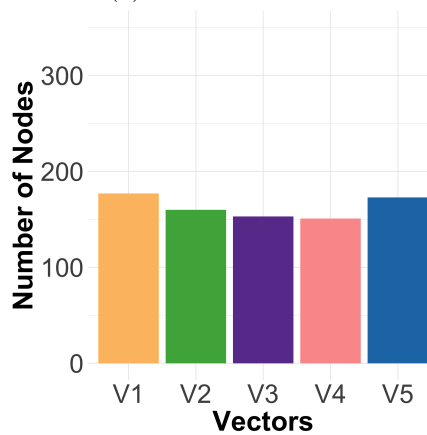
(c) MOEA/D on UF5.



(d) NSGA-II on UF5.



(e) MOEA/D on UF8.



(f) NSGA-II on UF8.

Fig. 4: The number of nodes for each vector found by MOEA/D is higher than those found by NSGA-II. MOEA/D seems to find more nodes via V1 and V2 vectors, showing a more unbalanced distribution while the distribution of nodes found by NSGA-II depends on the problem in question.

Table 3: *HV*, *IGD*, *Nodes*, Ratio of *Edges* given the number of nodes, Ratio of *Shared Nodes* given the number of nodes, Number of *Optimal Solutions*, *Components*, *Max* number of *in*-degrees, *Mean* number of *in*-degrees, *Max* number of *out*-degrees, *Mean* number of *out*-degrees of MOEA/D and NSGA-II.

MOEA/D-DE											
	HV	IGD	Nodes	Edges	Shared	Opt.	Comp.	Max in.	Mean in.	Max out.	Mean out.
UF1	0.86	0.01	578	2.64	0.72	0	3	10	3.04	10	3.04
UF2	0.86	0.01	693	2.40	0.65	0	3	10	2.53	10	2.53
UF3	0.57	0.17	465	3.04	0.76	6	1	23	3.77	23	3.77
UF4	0.47	0.04	593	2.44	0.61	0	3	18	2.96	18	2.96
UF5	0.63	0.04	529	2.76	0.70	4	3	13	3.32	13	3.31
UF6	0.64	0.00	646	2.47	0.63	0	3	10	2.72	10	2.72
UF7	0.70	0.01	679	2.35	0.62	0	3	13	2.58	13	2.58
UF8	0.66	0.07	1051	1.61	0.30	24	3	12	1.67	12	1.67
UF9	1.08	0.03	1022	1.62	0.28	1	3	10	1.72	10	1.72
UF10	0.07	0.42	839	1.95	0.38	0	3	13	2.09	13	2.09

NSGA-II											
	HV	IGD	Nodes	Edges	Shared	Opt.	Comp.	Max in.	Mean in.	Max out.	Mean out.
UF1	0.85	0.02	304	3.79	0.68	0	3	29	5.77	29	5.77
UF2	0.85	0.02	321	3.64	0.58	0	3	31	5.47	30	5.47
UF3	0.41	0.23	126	2.74	0.82	2	1	21	13.93	15	13.93
UF4	0.49	0.03	230	4.99	0.63	0	3	53	7.63	52	7.63
UF5	0.49	0.12	233	4.16	0.72	1	3	32	7.53	32	7.53
UF6	0.62	0.01	333	3.61	0.59	0	3	25	5.27	25	5.27
UF7	0.70	0.01	363	3.35	0.63	0	3	39	4.83	39	4.83
UF8	0.53	0.11	381	4.11	0.58	0	3	26	4.61	26	4.61
UF9	0.93	0.10	421	3.67	0.56	0	3	39	4.17	39	4.17
UF10	0.00	1.34	345	4.28	0.51	0	3	59	5.09	59	5.09

*force-directed* graph layout algorithms as implemented in R package *igraph* [?], that focuses on position the nodes to have few crossing of edges as possible.

In Figure 1, we can see the STN modeled for MOEA/D (a, on the top) and NSGA-II (b, on the bottom) on UF3. Before commenting on the results, we highlight that some vectors are not appearing in the images, and the reason is that they are following the same trajectory, reaching the nodes via the same paths. Looking at the STN for MOEA/D, we can see that all of the Pareto front (PF) optimal solutions, shown in red, in the trajectory found by V1, the light orange vector. The higher number of edges attributed to this component indicates that MOEA/D can reach the theoretical PF after exploring the region related to V1 in depth.

On the other hand, the STN of NSGA-II is much smaller, as we can also see in the number of nodes for each vector. It is no surprise that all the PF solutions found by NSGA-II are also at the end of the search since NSGA-II only accepts new solutions if they dominate old ones. Given the size of the red triangles, NSGA-II tends to come back to these solutions frequently.

Figure 2 shows that the STNs of MOEA/D and NSGA-II are bigger than the STNs modeled for UF3. We can see that the STN modeled for MOEA/D shows four optimal solutions found by different vectors during the search progress. After finding these solutions, MOEA/D continues exploring the problem until the end of the search. Now, NSGA-II also finds few optimal solutions, only one, but tends to come back to this solution frequently. We can see that the network components for both STN have little interaction with which other, suggesting that this problem poses more difficulties to both MOEA/D and NSGA-II than UF3.

Figure 3 illustrates the UF8 results. We can see that MOEA/D and NSGA-II conduct different searches over the three objective problems (as can also be seen in Table 3). Looking first at the results of MOEA/D, we can see that this algorithm progresses with the search with a higher exploration rate since, for all vectors, the number of edges and nodes is higher. Also, we can see that the STN of MOEA/D for the UF8 problem is denser than the NSGA-II for the same problem. MOEA/D performs the best in terms of the number of optimal solutions found in a problem, with 24 solutions in the theoretical Pareto front. By contrast, NSGA-II finds no optimal solution, and we can see that the network components are apart, suggesting that the starting point of the search poses a great impact on the search progress of NSGA-II.

Turning now to the STN metrics for each trajectory, in Figure 4, we can see that each of the MOEA/D vectors visit more nodes than the vectors of NSGA-II, meaning that MOEA/D finds more unique solutions than NSGA-II. This higher number of nodes, overall vectors, is not exclusive to the MOPs shown by Figure 4, as can be seen in Table 3. Moreover, these results, combined with the higher number of optimal solutions, lower values of maximum and mean in-/out-degrees, reinforce that MOEA/D can explore better the search space given the higher number of optimal solutions.

In terms of the traditional MOP metrics HV and IGD in Table 3, we can see that MOEA/D achieves higher HV values and lower IGD values, in comparison with NSGA-II for all MOPs. However, in UF7, both algorithms perform the same, and in UF4, NSGA-II performs the best in both metrics. Overall, for this set of MOPs, we consider that MOEA/D presents better performance than NSGA-II.

Table 3 also shows that the NSGA-II algorithm, for most problems, has more than three times more edges than nodes (MOEA/D keeps a lower ratio, with almost the same number of nodes and edges, with a ratio of edges around 2). This higher number of edges and nodes indicates that NSGA-II is visiting some nodes multiple times, which suggests that this metaheuristic might not be exploring the search space in depth during the search progress. A closer look at this Table also shows that the maximum out-degree value differs from the in-degree value in UF2, UF3, and UF4. This result indicates that NSGA-II visits the first node more than once, thus finding it challenging to find the right search trajectory direction to follow at the beginning of the search. Overall, these two results indicate that NSGA-II faces difficulties in some problems, not exploring many

new areas of the search space.

In summary, the STN models suggest an explanation for the differences in HV and IGD performance between MOEA/D and NSGA-II. The STNs showed that MOEA/D extensively explores the search space, finding more optimal solutions without re-visiting nodes too many times. On the other hand, the STNs modeled for NSGA-II indicate that this algorithm faces difficulties when exploring new areas of the search space and that its search progress depends on having good starting points. These results show that using STN for discriminating MOEAs is a viable method for analyzing and aggregating helpful information to the traditional practices for comparing such algorithms, such as the HV and IGD. A note of caution is due here since we only analyzed two MOEAs in a handful set of problems, with only two- and three objectives.

## 7 Conclusion

The main goal of the current study was to determine and design a methodology for applying the search trajectory networks (STNs) for modeling the search behavior of Multi-Objective Evolutionary Algorithms (MOEAs). The proposed method is based on the simple idea of extracting the STN features from some regions of the problem in question. We create STN models in a benchmark set that the most famous MOEAs, MOEA/D and NSGA-II, have different performances. Finally, we show that STNs can be effectively applied to differentiate visually and quantitatively.

Overall, this study strengthens the idea that characterizing and visualizing the search behavior of MOEAs can provide insightful advances into comprehending how distinct the search behaviors dynamics of such algorithms are and their overall performance. That is because we provided an STN modeling process that can discriminate well different MOEAs, aggregating helpful information to the traditional practices for comparing such algorithms, such as the HV and IGD. Thus, these findings have significant implications for understanding how different MOEAs perform when applied to solve multi-objective problems (MOPs). Moreover, we understand that this work shows excellent results of interest for the whole bio-inspired computation community.

A natural progression of this work is to investigate the generalization of the model to describe the search behavior of MOEAs other than MOEA/D and NSGA-II, especially in real-world and constrained MOPs. Furthermore, an analysis of the importance of features STN used and introduced in this work should be conducted in the context of automated landscape-aware selection and configuration MOEAs.

## References

1. Beume, N., Naujoks, B., Emmerich, M.: Sms-emoa: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research* 181(3), 1653–1669 (2007)

2. Bossek, J.: smooF: Single- and multi-objective optimization test functions. *The R Journal* (2017), <https://journal.r-project.org/archive/2017/RJ-2017-004/index.html>
3. Campelo, F., Aranha, C.: MOEADr: Component-wise MOEA/D implementation. URL <https://cran.r-project.org/package=MOEADr> (2018), r package version 1.2.0
4. Campelo, F., Batista, L., Aranha, C.: The MOEADr package: A component-based framework for multiobjective evolutionary algorithms based on decomposition. *Journal of Statistical Software* (2020), in press. Available from: <https://arxiv.org/abs/1807.06731>
5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6(2), 182–197 (2002)
6. Fieldsend, J.E., Alyahya, K.: Visualising the landscape of multi-objective problems using local optima networks. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. p. 1421–1429. GECCO '19, Association for Computing Machinery, New York, NY, USA (2019), <https://doi.org/10.1145/3319619.3326838>
7. Fonseca, C.M., Fleming, P.J.: On the performance assessment and comparison of stochastic multiobjective optimizers. In: Voigt, H.M., Ebeling, W., Rechenberg, I., Schwefel, H.P. (eds.) *Parallel Problem Solving from Nature — PPSN IV*. pp. 584–593. Springer Berlin Heidelberg, Berlin, Heidelberg (1996)
8. Kerschke, P., Grimme, C.: An expedition to multimodal multi-objective optimization landscapes. In: *Evolutionary Multi-Criterion Optimization, EMO*. Lecture Notes in Computer Science, vol. 10173, pp. 329–343. Springer (2017)
9. Li, H., Deb, K., Zhang, Q., Suganthan, P., Chen, L.: Comparison between MOEA/D and NSGA-III on a set of novel many and multi-objective benchmark problems with challenging difficulties. *Swarm and Evolutionary Computation* 46, 104–117 (2019)
10. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation* 13(2), 284–302 (2009)
11. Liefvooghe, A., Derbel, B., Verel, S., López-Ibáñez, M., Aguirre, H., Tanaka, K.: On pareto local optimal solutions networks. In: Auger, A., Fonseca, C.M., Lourenço, N., Machado, P., Paquete, L., Whitley, D. (eds.) *Parallel Problem Solving from Nature – PPSN XV*. pp. 232–244. Springer International Publishing, Cham (2018)
12. Miettinen, K.: *Nonlinear multiobjective optimization*, volume 12 of international series in operations research and management science (1999)
13. Ochoa, G., Tomassini, M., Verel, S., Verel, C.: A study of nk landscapes' basins and local optima networks. In: *Genetic and Evolutionary Computation Conference, GECCO*. pp. 555–562. ACM Press, New York, NY (2008)
14. Ochoa, G., Malan, K.M., Blum, C.: Search trajectory networks of population-based algorithms in continuous spaces. In: Castillo, P.A., Jiménez Laredo, J.L., Fernández de Vega, F. (eds.) *Applications of Evolutionary Computation*. pp. 70–85. Springer International Publishing, Cham (2020)
15. Ochoa, G., Malan, K.M., Blum, C.: Search trajectory networks: A tool for analysing and visualising the behaviour of metaheuristics. *Applied Soft Computing* 109, 107492 (2021), <https://www.sciencedirect.com/science/article/pii/S1568494621004154>

16. Schäpermeier, L., Grimme, C., Kerschke, P.: One plot to show them all: Visualization of efficient sets in multi-objective landscapes. In: Bäck, T., Preuss, M., Deutz, A., Wang, H., Doerr, C., Emmerich, M., Trautmann, H. (eds.) *Parallel Problem Solving from Nature – PPSN XVI*. pp. 154–167. Springer International Publishing, Cham (2020)
17. Schäpermeier, L., Grimme, C., Kerschke, P.: To boldly show what no one has seen before: A dashboard for visualizing multi-objective landscapes. In: *Evolutionary Multi-Criterion Optimization, EMO*. Lecture Notes in Computer Science, vol. 12654, pp. 632–644. Springer (2021)
18. Tsou, C.S.V.: Elitist non-dominated sorting genetic algorithm based on r. URL <https://cran.r-project.org/web/packages/nsga2R/nsga2R.pdf> (2013)
19. Tutar, T., Filipic, B.: Visualization of pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosection method. *IEEE Transactions on Evolutionary Computation* 19(2), 225–245 (Apr 2015), <https://doi.org/10.1109/tevc.2014.2313407>
20. Wang, R., Zhang, T., Guo, B.: An enhanced MOEA/D using uniform directions and a pre-organization procedure. In: *2013 IEEE Congress on Evolutionary Computation*. pp. 2390–2397 (2013)
21. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation* 11(6), 712–731 (2007)
22. Zhang, Q., Liu, W., Li, H.: The performance of a new version of MOEA/D on CEC’09 unconstrained MOP test instances. In: *Evolutionary Computation, 2009. CEC’09. IEEE Congress on*. pp. 203–208. IEEE (2009)
23. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W., Tiwari, S.: Multiobjective optimization test instances for the CEC 2009 special session and competition. University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report 264 (2008)