Decision/objective space trajectory networks for multi-objective combinatorial optimisation

Gabriela Ochoa, Arnaud Liefooghe, Yuri Lavinas, Claus Aranha

Decision/Objective Space Trajectory Networks for Multi-objective Combinatorial Optimisation

Gabriela Ochoa[1][0000−0001−7649−5669], Arnaud Liefooghe[2][0000−0003−3283−3122], Yuri Lavinas[3][0000−0003−2712−5340], and Claus Aranha[3][0000−0003−1390−7536]

[1] University of Stirling, Stirling, FK9 4LA, UK
gabriela.ochoa@stir.ac.uk
[2] Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
arnaud.liefooghe@univ-lille.fr
[3] University of Tsukuba, 1-1-1 Tennodai Tsukuba, 305-8577, Japan
lavinas.yuri.xp@alumni.tsukuba.ac.jp, caranha@cs.tsukuba.ac.jp

Abstract. This paper adapts a graph-based analysis and visualisation tool, search trajectory networks (STNs) to multi-objective combinatorial optimisation. We formally define multi-objective STNs and apply them to study the dynamics of two state-of-the-art multi-objective evolutionary algorithms: MOEA/D and NSGA2. In terms of benchmark, we consider two- and three-objective ρmnk-landscapes for constructing multi-objective multi-modal landscapes with objective correlation. We find that STN metrics and visualisation offer valuable insights into both problem structure and algorithm performance. Most previous visual tools in multi-objective optimisation consider the objective space only. Instead, our newly proposed tool asses algorithm behaviour in the decision and objective spaces simultaneously.

Keywords: algorithm analysis, search trajectory networks, STNs, combinatorial optimisation, visualisation, multi-objective optimisation

1 Introduction

Understanding the behaviour of search and optimisation algorithms remains a challenge to which visualisation techniques can contribute. The performance of multi-objective optimisation algorithms is usually visualised in the objective space, where the Pareto front (or an approximation of it) for two or three objectives is shown in a standard scatter plot; an idea that has been extended for more than three objectives using dimensionality reduction [21]. Incorporating the design space into the visualisation, however, can help to improve our understanding. Only a small number of approaches point in this direction, and most of them are tailored to continuous optimisation, such as cost landscapes [7], gradient field heatmaps [11], and the plot of landscapes with optimal trade-offs [20]. In the combinatorial domain, local optima networks [16, 19] have been adapted to multi-objective optimisation [6, 14]. These insightful visual approaches, however, concentrate on the fitness landscape structure, rather than on the algorithms dynamic behaviour.

The goal of this article is to adapt search trajectory networks (STNs) to multi-objective combinatorial optimisation. STNs were originally proposed for single-objective optimisation [17,18] as a graph-based tool to visualise and analyse the dynamics of any type of metaheuristic: evolutionary, swarm-based or single-point, on both continuous and discrete search spaces. STNs were later extended to multi-objective optimisation [12], but so far have been applied to continuous benchmark problems only. The extension of STNs from single- to multi-objective optimisation relies on the notion of *decomposition* [23], where the multi-objective problem is transformed into multiple single-objective scalar sub-problems. The idea is then to aggregate the STN of each these sub-problems to construct the multi-objective STN. One limitation of the approach proposed in [12] is that it considers a small number of decomposition vectors (5 to be precise), which restricts the granularity and expressing power of the modelling tool. In this paper, our contributions can be summarised as follows:

(1) We apply multi-objective STNs to combinatorial benchmarks, where both the landscape ruggedness and the correlation among objectives can be tuned.
(2) We offer a more formal definition of multi-objective STNs.
(3) We improve the granularity and accuracy of the modelling tool by increasing the number of decomposition vectors.
(4) We propose a 2D graph layout that conveys the design and objective spaces simultaneously in a single plot — this applies to two-objective problems only.

The paper is organised as follows. Section 2 introduces the necessary background on multi-objective optimisation. Section 3 formally defines the multi-objective STNs, together with the related metrics and visualisation techniques. Section 4 gives the experimental setup. Section 5 presents the experimental results of our analysis for both small and large multi-objective landscapes. At last, Section 6 concludes the paper and discusses further research.

2 Multi-objective Combinatorial Optimisation

This section provides definitions for multi-objective combinatorial optimisation, and presents two well-established multi-objective evolutionary algorithms.

2.1 Definitions

We assume an m-dimensional objective function vector $f\colon X \mapsto Z$ is to be maximised, such that every solution from the (discrete) solution space $x \in X$ maps to a vector in the objective space $z \in Z$, with $z = f(x)$ and $Z \subseteq \mathbb{R}^m$. Given two objective vectors $z, z' \in Z$, we say that z is dominated by z' if $z_i \leqslant z'_i$ for all $i \in \{1, \ldots, m\}$, and there is a $j \in \{1, \ldots, m\}$ such that $z_j < z'_j$. Similarly, a solution $x \in X$ is dominated by $x' \in X$ if $f(x)$ is dominated by $f(x')$. An objective vector $z^\star \in Z$ is non-dominated if there is no $z \in Z$ such that $z^\star$ is dominated by z. A solution $x^\star \in X$ is Pareto optimal if $f(x)$ is non-dominated. The set of Pareto optimal solutions is the Pareto set (PS), and its mapping

in the objective space is the Pareto front (PF). Evolutionary multi-objective optimisation (EMO) algorithms aim at identifying a PS approximation that is to be presented to the decision maker for further consideration [2, 4].

2.2 Multi-objective Evolutionary Algorithms

We consider two state-of-the-art EMO algorithms that are described below.

MOEA/D is a decomposition-based EMO algorithm that seek a high-quality solution in multiple regions of the objective space by decomposing the original (multi-objective) problem into a number of scalar (single-objective) sub-problems [23]. Let μ be the population size. A set $(\lambda^1, \dots, \lambda^i, \dots, \lambda^\mu)$ of uniformly-distributed weighting coefficient vectors defines the scalar sub-problems, and a population $P = (x^1, \dots, x^i, \dots, x^\mu)$ is maintained such that each solution x^i maps to the sub-problem defined by λ^i. Different scalarising functions can be used, and the weighted Chebyshev scalarising function [15] defined in the next section is a well-established example. A neighbourhood $\mathcal{B}(i)$ is additionally defined for each sub-problem $i \in \{1, \dots, \mu\}$, by considering its T closest weighting coefficient vectors. At each iteration, the population evolves with respect to a given sub-problem. Two solutions are selected at random from $\mathcal{B}(i)$ and an offspring is produced by means of variation operators. Then, for each neighbouring sub-problem $j \in \mathcal{B}(i)$, the offspring is used to replace the current solution x^j if there is an improvement in terms of the scalarising function. The algorithm iterates over sub-problems until a stopping condition is satisfied.

NSGA2 is an elitist dominance-based EMO algorithm using Pareto dominance for selection [5]. At a given iteration t, the current population P_t is merged with its offspring Q_t, and is divided into non-dominated fronts $\{F_1, F_2, \dots\}$ based on the non-dominated sorting procedure [9]. The front in which a given solution belongs to gives its rank within the population. Crowding distance is also calculated within each front. Selection is based on dominance ranking, and crowding distance is used as a tie breaker. Survival selection consists in filling the new population P_{t+1} with solutions having the best (smallest) ranks. In case a front F_i overfills the population size, the required number of solutions from F_i are chosen based on their crowding distance. Parent selection for reproduction consists of binary tournaments between randomly-chosen solutions, following the lexicographic order induced by ranks first, and crowding distance next.

3 Search Trajectory Networks

In order to define a graph-based model, we need to specify its nodes and edges. We start by giving these definitions for single-objective optimisation before describing how to construct the models for multiple objectives.

3.1 Definitions

Nodes are unique candidate solutions to the optimisation problem at each iteration, representing the status of the search process. In population-based algorithms, the best solution from the population (measured by the objective function) is typically chosen at each iteration as the representative solution. The set of nodes is denoted by N.

A Search trajectory is given by a sequence of representative solutions (nodes) in the order in which they are encountered during the search process.

Edges are directed and connect two consecutive nodes 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.

Single-objective STN model. An STN is a directed graph STN $= G(N, E)$ with nodes N and edges E as defined above. For constructing a single-objective STN, multiple runs of the algorithm under study are performed, and explored solutions and their transitions are aggregated into a single graph model. Notice that some solutions and transitions may appear multiple times during the sampling process. However, the graph retains as nodes each unique solution, and as edges each unique transitions among encountered solutions. Counters are maintained as attributes of the graph, indicating the frequency of occurrence of each (unique) node and edge.

ubsubsectionDecomposition-based STN sub-model.In multi-objective optimisation based on decomposition, the problem is decomposed into p scalar (single-objective) sub-problems that target different regions of the Pareto front [23]. A set of uniformly-generated weight vectors $\Lambda = (\lambda^1, \lambda^2, \ldots, \lambda^p)$ represents the scalar sub-problems defined by decomposition. For a given sub-problem $\lambda^j \in \Lambda$, the well-established Chebyshev scalarising function [15], to be minimised, is defined as follows:

$$g(x \mid \lambda^j) := \max_{i \in \{1,\ldots,m\}} \lambda_i^j \cdot \left| z_i^\star - f_i(x) \right| \tag{1}$$

such that $x \in X$ is a solution, $\lambda^j \in \mathbb{R}^m$ is a weighting coefficient vector and $z^\star \in \mathbb{R}^m$ is a reference point. The reference point is set to the best-known value for each objective.

In order to define the multi-objective STN, nodes are as described above, and edges separately follow the trajectories of each weight vector $\lambda^j \in \Lambda$, $j \in \{1, \ldots, p\}$. In other words, for a given sub-problem, the STN follows the trajectory of the solution with the best (lowest) Chebyshev scalar value for the corresponding weight vector. The trajectories for all weight vectors are then aggregated to construct a single graph model. Edges in the multi-objective STN are labelled by the vector whose transition they represent. The number of weight vectors p is a parameter of the modelling process. Section 5 reports the setting

Table 1: Description of STN metrics.

metric	description
nodes	number of unique solutions visited
pareto	number of solutions in the Pareto set
mean_pareto_in	average incoming degree to Pareto nodes
pareto_num_path	number of paths to Pareto nodes
pareto_mean_path	average shortest path to Pareto nodes

considered in our experiments. We offer below a more formal definition of the multi-objective STN model.

Multi-objective STN model ($\mathbf{STN_{MO}}$). Assuming we have $p = |\Lambda|$ single-objective sub-problems (weight vectors), the multi-objective STN model is obtained by the graph union of the p single-objective STNs. More formally, let $\text{STN}_{v1} = G(N_{v1}, E_{v1}), \text{STN}_{v2} = G(N_{v2}, E_{v2}), \ldots, \text{STN}_{vp} = G(N_{vp}, E_{vp})$ be the single-objective STNs for the sub-problems represented by vectors $(\lambda^1, \lambda^2, \ldots, \lambda^p)$, respectively. We construct STN_{MO} as the graph union of the STN_{vj} graphs, $j \in \{1, \ldots, p\}$. Specifically, $\text{STN}_{\text{MO}} = G(N_{v1} \cup N_{v2} \cup \ldots \cup N_{vp}, E_{v1} \cup E_{v2} \cup \ldots \cup E_{vp})$. The union graph contains the nodes and edges that are traversed for at least one of the weight vectors. Node and edge attributes indicate which weight vector(s) visited them.

3.2 Network Metrics

We introduce five network metrics to describe the behaviour of the algorithms. These metrics, summarised in Table 1, were selected as they have been found to relate to search performance in single-objective problems [18]. The number of nodes expresses the algorithm exploratory power, the number of Pareto optimal solutions indicates effectiveness, the mean incoming degree to Pareto nodes is reflective of how many trajectories were successful, the number of paths as well as the average shortest path to Pareto nodes are indicative of the algorithm efficiency in reaching Pareto optimal solutions.

3.3 Network Visualisation

Visualising networks is a powerful and often beautiful way of appreciating their structure, which can offer insights and even reflect features not easily captured by network metrics. Node-edge diagrams are the most familiar form of network visualisation, they assign nodes to points in the two-dimensional Euclidean space and connect adjacent nodes by straight lines or curves. Nodes and edges can be decorated with visual properties such as size, colour and shape to highlight important features.

Our proposed multi-objective STN visualisations (see Fig. 2 for an example we will analyse later), use node colours and shapes to identify four relevant types of nodes: (1) start of trajectories, (2) end of trajectories that do not reach a Pareto optimal solution, (3) intermediate solutions in the trajectories, and (4) solutions in the Pareto set. The size of nodes and the thickness of edges are proportional to their sampling frequency.

A key aspect of network visualisation is the graph-layout, which accounts for the positions of nodes in the 2D Euclidean space. Graphs are mathematical objects, they do not have a unique visual representation. Many graph-layout algorithms have been proposed. *Force-directed* layout algorithms, such as Fruchterman-Reignold [8], are based on physical analogies defining attracting and repelling forces among edges. They strive to satisfy generally accepted aesthetic criteria such as an even distribution of nodes on the plane, minimising edge crossings, and keeping a similar length of edges. We use force-directed layouts for visualising the multi-objective STNs with two and three objectives (Figs. 2, 3, 7). For two objectives, we additionally introduced a layout that takes advantage of the objective space. The idea is to use the two objective values as the nodes x and y coordinates (Figs. 4, 5, and 8). These plots allow us to appreciate the progression of the search trajectories in the design and objective spaces simultaneously. Our graph visualisations were produced using the igraph and ggraph packages of the R programming language.

4 Experimental Setup

This section describes the experimental setup of our analysis, including the considered benchmark problems as well as the parameters used for the STNs and for the algorithms.

4.1 Benchmark Problems

In terms of benchmark, we consider ρmnk-landscapes [22] for constructing multi-objective multi-modal landscapes with objective correlation. They extend single-objective nk-landscapes [10] and multi-objective nk-landscapes with independent objectives [1]. Candidate solutions are binary strings of size n. The objective function vector $f = (f_1, \ldots, f_i, \ldots, f_m)$ is defined as $f\colon \{0,1\}^n \mapsto [0,1]^m$ such that each objective f_i is to be maximised. The objective value $f_i(x)$ of a solution $x = (x_1, \ldots, x_j, \ldots, x_n)$ is an average value of the individual contributions associated with each variable x_j. Given objective f_i, $i \in \{1, \ldots, m\}$, and variable x_j, $j \in \{1, \ldots, n\}$, 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 $\{x_{j_1}, \ldots, x_{j_k}\}$. These f_{ij}-values are uniformly distributed in $[0,1]$. Thus, the individual contribution of a variable x_j depends on its own value and on the values of $k < n$ variables other than x_j.

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,

Table 2: Benchmark parameters for small and large ρmnk-landscapes.

description	**values**
number of variables	$n = 16$ (small), $n = 128$ (large)
number of interactions	$k \in \{1, 4\}$
number of objectives	$m \in \{2, 3\}$
objective correlation	$\rho \in \{-0.4, 0.0, 0.4\}$

following the random model from [10]. By increasing the number of variable interactions k, landscapes can be gradually tuned from smooth to rugged. In ρmnk-landscapes, f_{ij}-values additionally 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 $p, q \in \{1, \ldots, m\}$ with $p \neq q$, where $\rho > \frac{-1}{m-1}$ defines the correlation among the objectives; see [22] for details. The positive (resp. negative) correlation ρ decreases (resp. increases) the degree of conflict between the objective values.

Interestingly, ρmnk-landscapes exhibit different characteristics and degrees of difficulty for EMO algorithms [3,13].The source code of the ρmnk-landscapes generator is available at the following URL: http://mocobench.sf.net.

4.2 Parameter Setting

We generate 12 small and 12 large ρmnk-landscapes with the parameter settings listed in Table 2. This allows us to investigate the differences between small and large instances, two and three objectives, conflicting, independent or correlated objectives, all this for relatively smooth to relatively rugged landscapes.

In terms of algorithms, we experiment with both MOEA/D and NSGA2 under the parameters from Table 3. Each algorithm is run independently 10 times on each instance. Algorithm performance is given in terms of hypervolume [24]. More particularly, we measure the relative hypervolume deviation with respect to the exact PF (for small instances) or best-known PF (for large instances). Let hv be the hypervolume covered by the population, the relative hypervolume deviation is $(hv^\star - hv)/hv^\star$, such that $hv^\star$ is the best-known hypervolume. A lower value is thus better. The hypervolume reference point is set to the origin.

Table 3: Algorithm parameters for MOEA/D and NSGA2.

description	**values**
population size	$\mu = 101$ ($m = 2$), $\mu = 231$ ($m = 3$)
neighbourhood size	$T = 10$ (MOEA/D)
variation	1-point crossover, bit-flip mutation with rate $1/n$
number of generations	$g = 20$ ($n = 16$), $g = 500$ ($n = 128$)

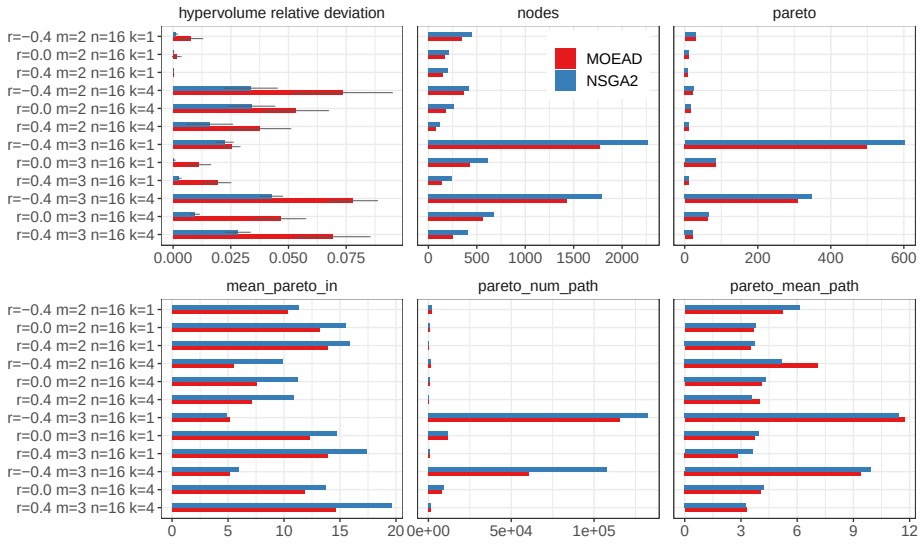

Fig. 1: Algorithm performance and STN metrics for *small* instances.

4.3 Reproducibility

For reproducibility purposes, relevant data and code are be available at: https://github.com/gabro8a/STNs-MOCO

5 Results

This section reports and comments the STNs obtained for small instances, and then for large instances. STN metrics are also discussed and related with algorithm performance.

5.1 Small Instances

We start with results for *small* instances with $n = 16$. In this case, the STN modelling used $p = 101$ decomposition vectors for instances with two objectives and $p = 231$ vectors for instances with three objectives; i.e. the same setting as the algorithms' population size. This give us the maximum possible modelling granularity (one vector per each individual member in the population), while still producing interpretable images.

Network Metrics. Algorithm performance for the 12 small instances is reported in Fig. 1 (top left), together with the five network metrics described in Table 1. For this set of instances, NSGA2 consistently outperforms MOEA/D, as indicated by the lower hypervolume relative deviation values. The higher STN

metric values obtained by NSGA2 for nodes, pareto and pareto_num_path clearly support this trend. Another clear trend from the STN metrics is the decrease in values when we go from conflicting ($\rho = -0.4$) to positively correlated objectives ($\rho = 0.4$), which is observed for both values of $k \in \{1, 4\}$ and $m \in \{2, 3\}$. Finally, a salient observation from Fig. 1 is the large metric values observed for instances with $m = 3$ and conflicting objectives ($\rho = -0.4$). The values of nodes, pareto, pareto_num_path and pareto_mean_path are higher for $k = 1$ than for $k = 4$. This is consistent with previous findings: although there are more local optima for larger k values, the number of global optima (i.e. Pareto optimal solutions) has the opposite trend and decreases with increasing k [22].

Network Visualisation with a Force-Directed Layout. Figs. 2 and 3 provide examples using a force-directed layout for $m = 2$ and $m = 3$ objectives, respectively. They report the multi-objective STN obtained for MOEA/D (top) and NSGA2 (bottom) for conflicting (left), independent (middle) and correlated objectives (right). The network visualisations confirm the trends observed in the metrics. Notably, the number of nodes in the networks consistently decreases when moving from negatively correlated objectives ($\rho = -0.4$, left) to positively correlated objectives ($\rho = 0.4$, right). We can also visually confirm the much denser STNs obtained for $m = 3$ objectives, as reported in Fig. 3.

Network Visualisation with the Objective-Space Layout. Figs. 4 and 5 shows our proposed objective-space network layouts applied to the two studied levels of ruggedness $k \in \{1, 4\}$, respectively. Notice that this layout is only applicable for $m = 2$ objectives if we restrict ourselves to the 2D Euclidean space. We argue that this layout may be more useful to the multi-objective optimisation community (as compared to the force-directed layouts shown in Figs. 2 and 3) as they resemble the familiar Pareto front scatter plots. However, they offer additional insights, revealing not only the Pareto front when it is reached, but also the search progress towards it, thus giving indication of unsuccessful runs as well. Notice that in these plots, an additional graphical layer is shown in the form of blue diamonds. They correspond to the exact Pareto front and are not part of the STN nodes. They serve as a tool to appreciate if and when the STN trajectories reach the Pareto front. The objective-space layout, therefore, might be more suitable for appreciating the performance difference between algorithms.

With respect to NSGA2 outperforming MOEA/D (as indicated by the performance metric in Fig. 1), this can only be clearly appreciated for $k = 4$ and $m = 2$ (Fig. 5). Looking at the left plots for $\rho = -0.4$, we can confirm that the MOEA/D STN (top plot) has a larger number end nodes (orange triangles) that are also of larger size as those of the NSGA2 STN (bottom plot). Remember that the size of nodes is proportional to their sampling frequency. Therefore, this is a visual reflection that MOEA/D has a larger number of unsuccessful runs, that is, trajectories ending into sub-optimal solutions. The NSGA2 STN (bottom plot) reveals in this case a larger number of red nodes (Pareto solutions), which are of larger size. Remember that the the super-imposed blue diamond scatter

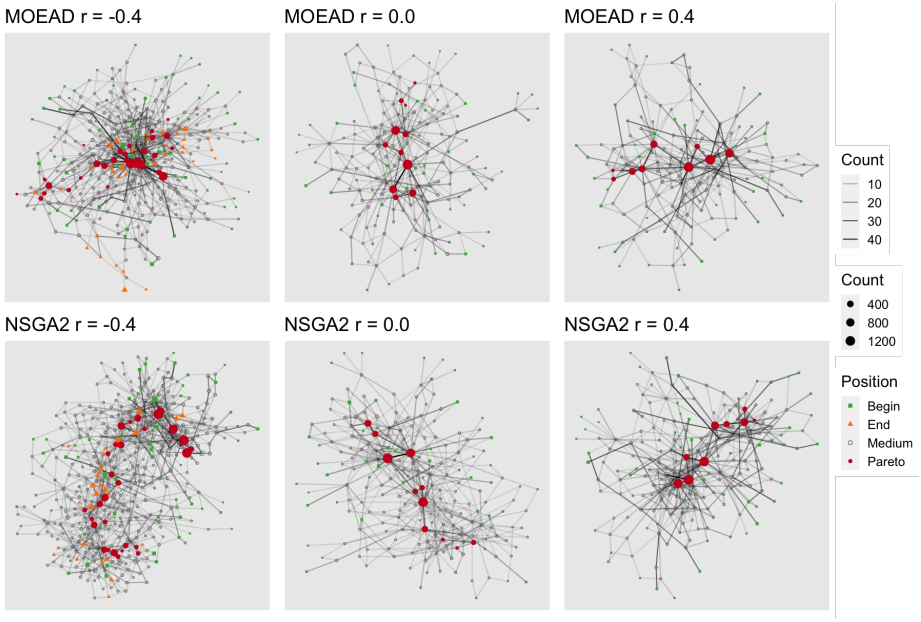

Fig. 2: STN visualisation with a force-directed layout for *small* instances with $k = 1$ and $m = 2$ objectives.

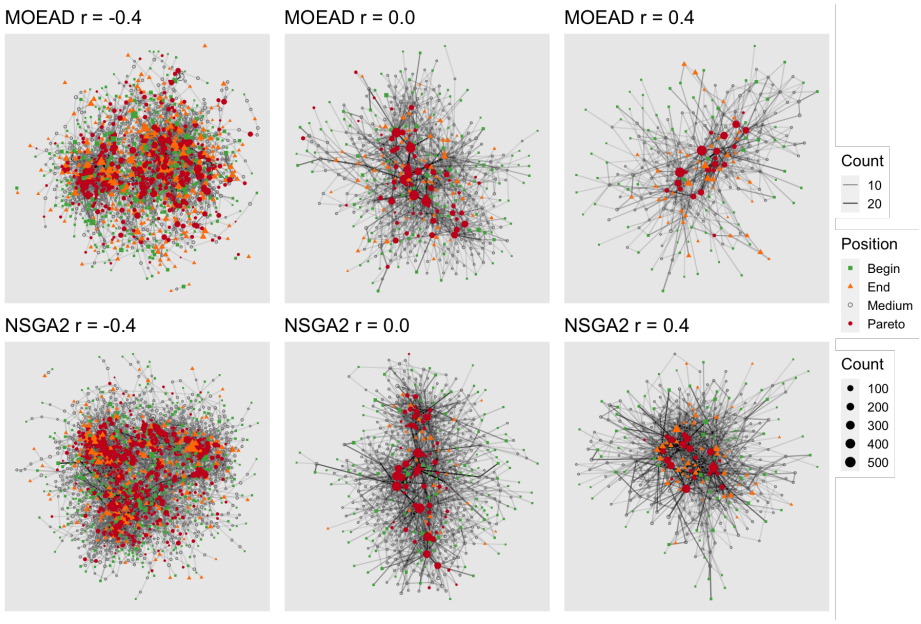

Fig. 3: STN visualisation with a force-directed layout for *small* instances with $k = 4$ and $m = 3$ objectives.

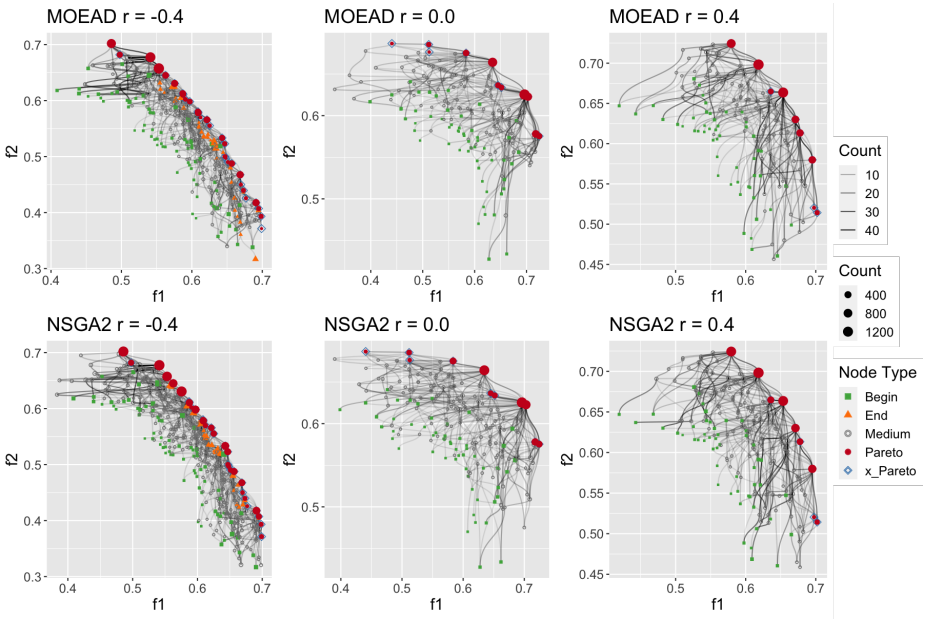

Fig. 4: STN visualisation with the objective-space layout for *small* instances with $k = 1$ and $m = 2$ objectives.

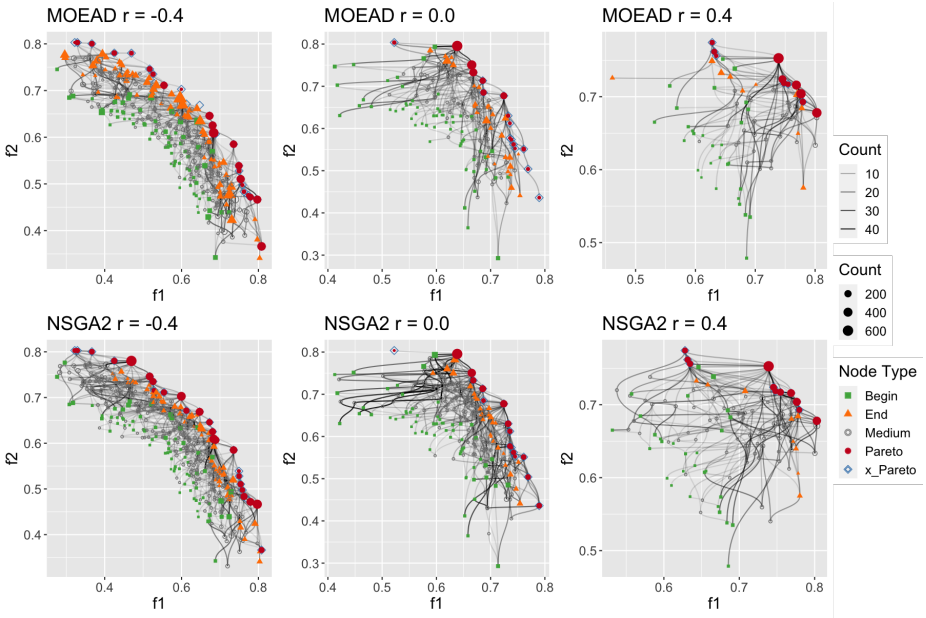

Fig. 5: STN visualisation with the objective-space layout for *small* instances with $k = 4$ and $m = 2$ objectives.

plot is used to visually locate the exact Pareto front. A careful inspection of the MOEA/D STN (top plot in Fig.5) reveals one empty blue diamond, and some diamonds that are only partially filled with red nodes (Pareto solutions found by the trajectories).

5.2 Large Instances

We continue our discussion by analysing the results for *large* instances with $n = 128$. The STN modelling used $p = 51$ decomposition vectors for instances with two objectives and $p = 66$ vectors for instances with three objectives. In this case, we used fewer weight vectors relative to the population size for efficiency reasons, and for improving both the cosmetic rendering and interpretability of the STN images. Notice that the larger the number of vectors, the larger the number of nodes in the STN models. The number of vectors can be seen as a parameter to adjust the model granularity.

Network Metrics. Algorithm performance and network metrics for large instances are reported in Fig. 6. For this set of instances, there is less difference in performance between the two algorithms. Nevertheless, notable exceptions appear for $m = 3$ objectives and conflicting objectives ($\rho = -0.4$), where MOEA/D reaches significantly better hypervolume values. This is supported by the higher STN metric values obtained by MOEA/D for nodes, pareto and pareto_num_path on the corresponding instances. Notice that for $\rho = 0.4$, $m = 2$, $k = 4$ in Fig. 6, NSGA2 does not find Pareto optimal solutions, therefore, some of the metrics cannot be computed, which explains the absence of the blue bar in this case. We notice that the NSGA STNs contain much more nodes, which is to be contrasted by its number of pareto nodes that is often particularly low compared to MOEA/D. This suggests that NSGA2 has a higher rate of discovery, but that it gets more easily trapped into sub-optimal solutions.

Network Visualisation with a Force-Directed Layout. We report in Fig. 7 examples of multi-objective STNs using a force-directed layout, for $m = 3$ objectives and $k = 4$. We observe that the networks are much denser than for small instances, although we used comparatively fewer decomposition vectors. This is to be expected given the exponentially larger search space of large instances. For conflicting objectives ($\rho = -0.4$, left), MOEA/D significantly outperforms NSGA2. We observe that MOEA/D identifies significantly more Pareto optimal solutions, which confirms the trend observed in the STN metrics. For uncorrelated objectives ($\rho = 0.0$, middle), the NSGA2 STN contains fewer Pareto optimal solutions than for MOEA/D, but they are identified more frequently, given the size of pareto nodes (in red). At last, for positively correlated objectives ($\rho = 0.4$, right), both algorithms identify about the same number of Pareto optimal solutions, but we still see that NSGA2 identifies them more frequently, which supports the fact that NSGA2 is slightly better for this instance.

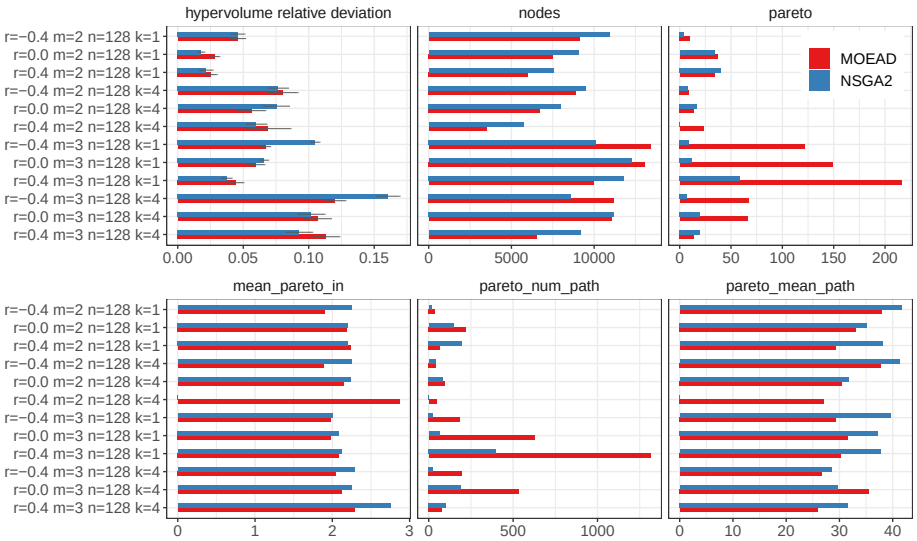

Fig. 6: Algorithm performance and STN metrics for *large* instances.

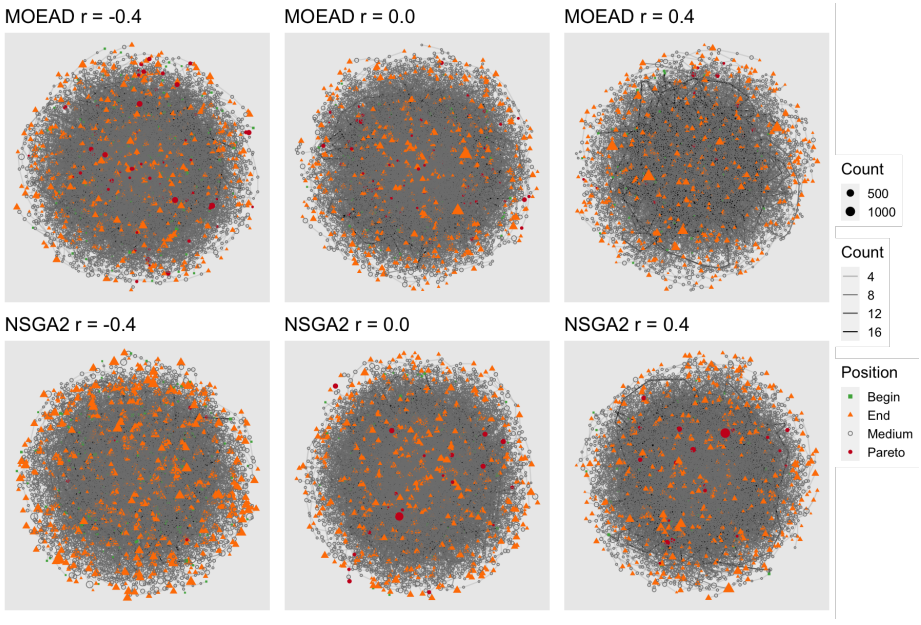

Fig. 7: STN visualisation with a force-directed layout for *large* instances with $k = 4$ and $m = 3$ objectives.

Network Visualisation with the Objective-Space Layout. Let us now analyse the objective-space network layout for large instances with $m = 2$ ob-

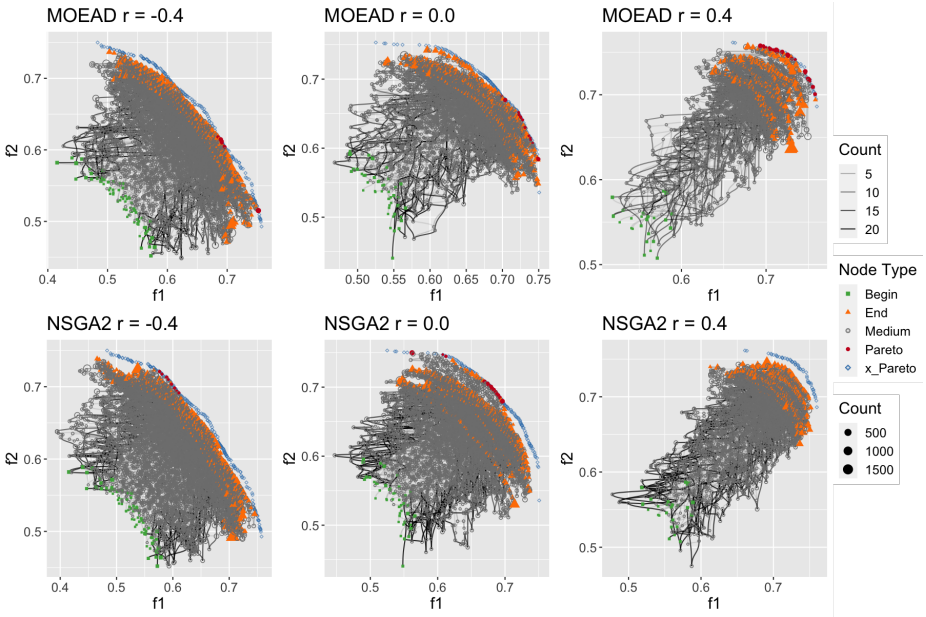

Fig. 8: STN visualisation with the objective-space layout for *large* instances with $k = 4$ and $m = 2$ objectives.

jectives. The multi-objective STNs are reported in Fig. 8 for $k = 4$. The visualisations for $k = 1$ are not shown due to space constraints, but they show similar trends. As anticipated by the analysis of STN metrics in Fig. 6, the multi-objective STNs obtained by the two algorithms are similar for the large two-objective instance, although solutions seem a bit more spread for NSGA2. The number of pareto nodes tends to be proportionally higher as we gradually shift from conflicting ($\rho = -0.4$) to positively correlated objectives ($\rho = 0.4$). A notable difference is for NSGA2 and $k = 4$, where the STN contains no pareto nodes for positively correlated objectives ($\rho = 0.4$). Furthermore, the position of end nodes (orange triangles) suggests that the trajectories end up in sub-optimal solutions farther away from the Pareto front for instances with $k = 4$, for which there are more local optima. Going back to the comparison between MOEA/D and NSGA2, the objective-space network layout of the STNs provide visual evidence confirming that, although NSGA2 seems to explore more solutions, it is attracted to lower quality solutions.

6 Conclusions

We argue that STNs are an accessible tool to analyse and visualise the behaviour of evolutionary multi-objective optimisation algorithms. Constructing STN models does not require any specific sampling techniques. Instead, data is collected

from a set of runs of the studied algorithms, and then aggregated and processed to devise the models. Post-processing tools, however might be required to deal with large models. STNs provide insights into problem structure as well as into algorithm convergence behaviour and performance differences.

Future work could study additional multi-objective problems and algorithms, including real-world problems and 4+ objectives. The challenge we foresee here deals with the larger number of solutions attained by the trajectories. For this, we could thoroughly investigate coarser models including varying the number of decomposition vectors in the STN model, and of grouping multiple solutions within a single node, as has been done for single-objective STN models [17, 18]. A number of repositories contain code and data to start with STN modelling and analysis for both single-objective[4] (including a web-based tool[5]), and multi-objective combinatorial[6] and continuous [7] problems. We should provide unified software tools to improve the usability of STN models.

Acknowledgements

We are deeply grateful to the SPECIES Society for funding a scholarship for Yuri Lavinas to visit the University of Stirling, Scotland, UK.

References

1. Aguirre, H.E., Tanaka, K.: Working principles, behavior, and performance of MOEAs on MNK-landscapes. European Journal of Operational Research 181(3), 1670–1690 (2007)
2. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems. Springer, New York, NY (2007)
3. Daolio, F., Liefooghe, A., Verel, S., Aguirre, H., Tanaka, K.: Problem features versus algorithm performance on rugged multiobjective combinatorial fitness landscapes. Evolutionary Computation 25(4), 555–585 (2017)
4. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. Wiley, Chichester, UK (2001)
5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multi-objective 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)
7. Fonseca, C.M., Fleming, P.J.: On the performance assessment and comparison of stochastic multiobjective optimizers. In: Parallel Problem Solving from Nature, PPSN IV. pp. 584–593. Springer Berlin Heidelberg, Berlin, Heidelberg (1996)

[4] https://github.com/gabro8a/STNs
[5] http://45.32.184.82
[6] https://github.com/gabro8a/STNs-MOCO
[7] https://github.com/gabro8a/STNs-MOEA

8. Fruchterman, T.M.J., Reingold, E.M.: Graph drawing by force-directed placement. Softw. Pract. Exper. 21(11), 1129–1164 (Nov 1991)
9. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Boston, MA, USA (1989)
10. Kauffman, S.A.: The Origins of Order. Oxford University Press (1993)
11. 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)
12. Lavinas, Y.C., Aranha, C., Ochoa, G.: Search trajectories networks of multiobjective evolutionary algorithms. In: Applications of Evolutionary Computation, EvoApplications. Lecture Notes in Computer Science, vol. 13224, pp. 223–238. Springer (2022)
13. Liefooghe, A., Daolio, F., Verel, S., Derbel, B., Aguirre, H., Tanaka, K.: Landscape-aware performance prediction for evolutionary multi-objective optimization. IEEE Transactions on Evolutionary Computation 24(6), 1063–1077 (2020)
14. Liefooghe, 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)
15. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers (1999)
16. 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)
17. 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)
18. 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)
19. Ochoa, G., Veerapen, N., Daolio, F., Tomassini, M.: Understanding phase transitions with local optima networks: Number partitioning as a case study. In: Evolutionary Computation in Combinatorial Optimization, EvoCOP. Lecture Notes in Computer Science, vol. 10197, pp. 233–248 (2017)
20. Schäpermeier, L., Grimme, C., Kerschke, P.: One plot to show them all: Visualization of efficient sets in multi-objective landscapes. In: Parallel Problem Solving from Nature, PPSN XVI. pp. 154–167. Springer International Publishing, Cham (2020)
21. Tušar, T., Filipič, 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 (2015)
22. Verel, S., Liefooghe, A., Jourdan, L., Dhaenens, C.: On the structure of multi-objective combinatorial search space: MNK-landscapes with correlated objectives. European Journal of Operational Research 227(2), 331–342 (2013)
23. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation 11(6), 712–731 (2007)
24. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Da Fonseca, V.G.: Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation 7(2), 117–132 (2003)