This is a post-peer-review, pre-copyedit version of an article published in Liefooghe A & Paquete L (eds.) *Evolutionary Computation in Combinatorial Optimization*. Lecture Notes in Computer Science, 11452. The 19th European Conference on Evolutionary Computation in Combinatorial Optimisation, Leipzig, Germany, 24.04.2019-26.04.2019. Cham, Switzerland: Springer International Publishing, pp. 163-178. The final authenticated version is available online at: https://doi.org/10.1007/978-3-030-16711-0_11

Clarifying the Difference in Local Optima Network Sampling Algorithms

Sarah L. Thomson[1] 0000-0001-6971-7817, Gabriela Ochoa[1] 0000-0001-7649-5669, and Sébastien Verel[2] 0000-0003-1661-4093

[1] Computing Science and Mathematics, University of Stirling, Stirling, UK
{s.l.thomson,gabriela.ochoa}@stir.ac.uk
[2] Université du Littoral Côte d'Opale, EA 4491 - LISIC, Calais, France
verel@uni-littoral.fr

Abstract. We conduct the first ever statistical comparison between two *Local Optima Network* (LON) sampling algorithms. These methodologies attempt to capture the connectivity in the local optima space of a fitness landscape. One sampling algorithm is based on a random-walk *snowballing* procedure, while the other is centred around multiple traced runs of an Iterated Local Search. Both of these are proposed for the Quadratic Assignment Problem (QAP), making this the focus of our study. It is important to note the sampling algorithm frameworks could easily be modified for other domains. In our study descriptive statistics for the obtained search space samples are contrasted and commented on. The LON features are also used in linear mixed models and random forest regression for predicting heuristic optimisation performance of two prominent heuristics for the QAP on the underlying combinatorial problems. The model results are then used to make deductions about the sampling algorithms' utility. We also propose a specific set of LON metrics for use in future predictive models alongside previously-proposed network metrics, demonstrating the payoff in doing so.

Keywords: Combinatorial Fitness Landscapes, Local Optima Networks, Quadratic Assignment Problem

1 Introduction

Local optima networks (LON) are a partial fitness landscape of a combinatorial optimisation problem [1]. Features of the networks have repeatedly been linked to heuristic search in various famous problem domains ([2–5]). LONs were originally used for fully-enumerated fitness landscapes and therefore reflected small problems. More recently, sampling algorithms have been proposed [5–7] with the aim of using the analysis on larger problems. The nature, biases and resultant samples of the algorithms are of critical importance in moving LON analysis towards real-world systems. In this study we conduct the first comparison of sampling algorithms for this purpose. Features of the samples are contrasted, and the features are used in linear and random forest models for predicting heuristic performance on the underlying combinatorial problems. In other words, we examine which of the two most recent sampling algorithms ([6, 7]) produces LONs with more predictive power, in terms of their ability to measure search difficulty. We also propose using funnel metrics as features for predictive models of algorithm performance. A *funnel* is a large fitness landscape feature and is essentially a basin of attraction at the level of local optima. The main contributions of this paper are:

1. The first descriptive comparison of local optima network sampling algorithms.
2. Contrast of the differences in predictive power of the LON features for explaining meta-heuristic variance.
3. The addition of a specific funnel metric set for predicting algorithm performance in LON models using linear and random forest regression.

2 Definitions

A fitness landscape [8] is a triplet (S, N, f) where S is the set of all possible solutions, $N : S \longrightarrow 2^S$, a neighbourhood structure, is a function that assigns to every $s \in S$ a set of neighbours $N(s)$, and f is a fitness (objective value) function such that f : S $\longrightarrow \mathbb{R}$, where the fitness value is a real number that can be viewed as the *height* of a given solution in the landscape.

A *local optima network* is a representation of the fitness landscape at the level of local optima. We now formally define the constituent parts of a local optima network, before proceeding to describe the object as a whole.

Nodes. The set of nodes, LO, is comprised of local optima, i.e. a solution lo_i satisfies the condition that it has superior fitness to all other solutions in its neighbourhood: $\forall n \in N(lo_i) : f(lo_i) \geq f(n)$, where $N(lo_i)$ is the neighbourhood and n is a single neighbour.

Edges. The set of edges, E, consists of directed and weighted links. An edge is traced if the probability of 'escape' — using perturbation and then hill-climbing — from the source node to the destination is greater than zero, and is weighted with the probability. Formally, local optima lo_i and lo_j form the source and destination of an edge iff $w_{ij} > 0$.

Local optima network (LON). The weighted local optima network $LON = (LO, E)$ is a graph where the nodes $lo_i \in LO$ are the local optima, and there exists an edge $e_{ij} \in E$, with weight w_{ij}, between two nodes lo_i and lo_j if $w_{ij} > 0$. Note that w_{ij} may be different than w_{ji}. Thus, two weights are needed in general, and so a local optima network is an oriented transition graph.

2.1 The 'Network' Feature Set

We now describe and introduce terminologies for the complex network features we use in this study. The number of edges found in one of our sampled networks is referred to as *edges*. We also use the mean fitness found in the sample of local optima comprising the network nodes, and denote this as *meanfitness*. The network diameter (the longest possible length in number of edges separating two nodes), *diam*, and the average out-degree (the number of edges which are directed away from a particular node), *outdegree*, are also included.

2.2 The 'Funnel' Feature Set

A fitness landscape *funnel* is a basin of attraction at the level of local optima. Instead of a single local optimum as an attractor for surrounding solutions, every member of a funnel is a local optimum. Each of these is on a path which, if followed, terminates at a single high-quality local optimum — the bottom of the funnel. These funnel bottoms are also referred to as *sinks* from a graph theory perspective, and are simply the nodes with no outgoing edges. We uncover the funnels by first pruning off any LON edges which are non-improving in fitness, and then by commencing a depth-first search from a funnel bottom, in the process revealing the sequences of local optima which must terminate there.

We use four features relating to fitness landscape funnels in our analysis here. The first is simply the amount of funnels as a proportion of the total node count, *funnel*. We also use the number of compressed local optima. The compression process happens during the network pre-processing for funnel analysis, where the non-improving edges are removed, and then plateaus at the local optima level are joined into a single node. The metric *ncoptima* is the number of optima left after this processing of the network. Also considered is the incoming flow to sub-optimal funnel bottoms or sinks (which can be calculated by the addition of the incoming edge-weights to the sink nodes) — this is referred to as *substrength*, and it comes as a proportion of the total flow going to all funnel bottoms, including optimal ones. The fourth funnel feature is *sinkfitness*, which is simply the mean fitness of funnel bottom nodes (sinks in the network).

3 Experimental Setting

3.1 Benchmark Test Problem

Our test instances are Quadratic Assignment Problems (QAP), where an optimal assignment of N facilities to N locations is sought. Each facility, location pair has an associated distance and flow between them. An instance of QAP, then, is defined with two matrices: a *distances* matrix, and a *flow* matrix. A solution to the QAP, π, is a permutation of facilities, while the objective value is calculated as follows:

$$C(\pi) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} b_{\pi_i \pi_j}$$

where $A = \{a_{ij}\}$ and $B = \{b_{ij}\}$ are the distance and flow matrices, respectively.

We focus exclusively on the much-studied Quadratic Assignment Problem Library (QAPLIB). QAPLIB boasts a diverse selection of problems, both synthetic and real-world, from different classes. Here we use 30 moderately-sized instances, from various problem classes. The problem dimensions N are between 25 and 50, meaning 25-50 facilities to be assigned. Occurrences of the QAP can largely be placed in one of four categories: Uniform random distances and flows; random flows on grids; real-world problems; and random real-world like problems. Instances from all of these categories are considered.

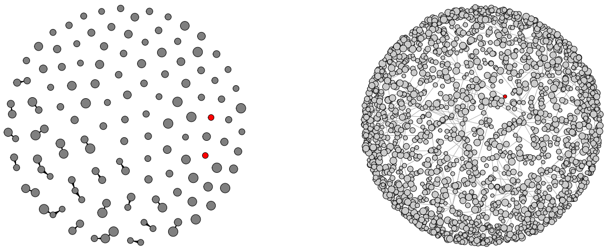
Uniform random distances and flows. This class of QAP is known to be difficult for optimisation heuristics. The entries for both the distance and flow matrices are taken at random from a Gaussian distribution. The naming convention for them in the QAPLIB is *tainna*, with *nn* being the problem dimension. The nature of the points on the plane is random. We use here *tai25a*, *tai30a*, *tai35a*, and *tai40a*.

Random flows on grids. For these problems, the locations are each situated in one square on an $m{*}n$ grid, which is rectangular. The flow matrix entries are generated randomly. From this category, we use *nug25*, *nug27*, *nug28*, *nug30*, *sko42*, *tho30*, *tho40*, and *wil50*.

Real-world problems. These problems arise from practical applications. We briefly describe the instances used in this article. We use two *bur* instances, which comprise of the average stenotypist's typing data. Also included are some *kra* problems: these were used to plan a German hospital. The *esc* instance we use presents itself in computer science (the testing of sequential circuits). In the *ste* problem set, the goal is to minimise the length of connections between units that have to be placed on a rectangular grid. The real-world instances we use in our analysis are *bur26a*, *bur26b*, *esc32e*, *kra30a*, *kra30b*, *kra32*, *lipa30a*, *lipa30b*, *lipa40a*, *lipa40b*, *ste36a*, *ste36b*, and *ste36c*.

Random real-world like problems. The naming convention is *tainnb*, where *nn* is the problem dimension. We use the problems *tai25b*, *tai30b*, *tai35b*, and *tai40b*.

Miscellaneous. The *chr* set of problems are a special occurrence of the QAP and do not fit into any of the previous categories. The flow matrix forms a mathematical *tree*, with no particular specification for the definition of the distance matrix. We use *chr25a* from this set.



(a) *optSample* LON sample for instance *wil50*. Top 5% of optima by fitness.

(b) *walkSample* LON sample for instance *wil50*. Top 0.5% of optima by fitness.

Fig. 1. A representative LON for an instance from the Quadratic Assignment Problem Library. Figures show LON samples as obtained by two different sampling methods. The global optimum or optima in the samples are shown in red.

3.2 The Sampling Algorithms

Snowball (Chain-referral) Sampling. Snowball sampling is a process, originating in the social sciences, where each respondent in a survey asks a few of their friends to complete the survey as well. The social network of survey respondents is then analogous to an ever-growing rolling snowball, growing larger with every step. This was introduced as the mechanism behind a sampling algorithm for local optima networks very recently [7].

To construct the local optima network sample, the algorithm first starts from a random solution and hill-climbs to a local optimum, x_t. This local optimum is the start-point of the walk. Then the recursive snowball procedure branches from x_t: the neighbouring local optima are explored, and then the neighbours of those are explored, up to a specified 'depth' given by a parameter d. The local optima are found by carrying out mutations followed by hill-climbing. The hill-climbing used to obtain local optima is best-improvement and uses a random pairwise swap of facilities as the operator. The perturbation operator is four of these mutations in a row, i.e. four swaps.

After the snowball expansion of x_t, the algorithm returns to the initial local optimum x_t and continues on its walk from there. Specifically, a neighbouring local optimum *which is not already present in the walk* is found. If all neighbouring local optima have been expanded, a new random solution is generated and hill-climbing applied to find a local optimum for inclusion in the random walk. Then, that optimum is included in the walk as x_{t+1} and becomes the 'centre' from which a new iteration of snowballing expansion begins. The algorithm then goes back x_{t+1} and finds a neighbour to be x_{t+2} on the walk, and so on. The walk $x_{t...t+n}$ is of a length given by the sampling parameter l.

LON snowball sampling is configurable with three parameters: l (length of random walk), d (depth of snowballing), and m (number of sampled edges). We use the three (l, d) combinations suggested in [7] for obtaining our LONs: $(100, 60)$, $(400, 30)$, and $(100, 30)$, alongside the suggested value for m, which is two. The parameters l and d can be used to tune the sample obtained.

A full description of LON snowballing, including pseudo-code and descriptive figures, can be found in [7]. Algorithm 2 is particularly helpful. In the text that follows, we refer to this sampling algorithm using the term *walkSample*, to reflect the fact it is a sample based on a random walk.

Iterated Local Search Sampling. This sampling method is built around the competent *Improved Iterated Local Search* [9] heuristic for the QAP, and was proposed in [6]. The local optima network is built while the heuristic algorithm is attempting optimisation. In essence, 200 ILS runs are commenced from different random solutions; for each run, as the ILS progresses in the minimisation process, all local optima encountered are recorded. Alongside these, every transition between two local optima (using perturbation and then hill-climbing) is saved as an edge in the network (LON), or if the edge has already been recorded, the weight of the edge is incremented. Each run terminates either when the global optimum is found, or after 10000 iterations without improvement.

The local search stage uses a first improvement hill-climbing variant with a random pairwise swap operator (the same as the LS in *walkSample*. The perturbation operator exchanges k randomly chosen items. In the original ILS algorithm code, a few settings for k are suggested. We use a subset of those in this study, namely: $\frac{n}{4}$, $\frac{n}{2}$, and $\frac{3n}{4}$. These settings are important for the construction of the LONs, because the perturbation mechanism controls the discovery and inclusion of network edges (connections between nodes). A comprehensive description of both the original ILS and the modified version for LON sampling can be found in [6]. We refer to this sampling process henceforth by *optSample*, to reflect the fact that the sampling is done alongside heurisitic optimisation.

Figure 1 compares a sample from *optSample* algorithm with one obtained using *walkSample*. The Figure serves as a precursor to a formal comparative analysis later on, but captures a remarkable amount of information. Both are sampled local optima networks, extracted from the same Quadratic Assignment Problem. The sample in Figure 1a is extracted using *optSample*, while *walkSample* was used to obtain the network in Figure 1b.

In red is the global optimum (or the global optima), while all other nodes are grey in colour. Both samples have been capped by fitness: only optima in the top 5% (Figure 1a) and 0.05% (Figure 1b) of the fitness distributions are included in the plots. The latter percentile is lower because the sample had orders of magnitude more optima and edges crowding its network at this fitness level.

Comparing the two plots, there are striking differences. Taking them in turn, we can see that the high-quality sub-LON which *optSample* produces (Figure 1a) is extremely sparse, and indeed the two global optima are disconnected (isolated) nodes in this graph. This could indicate that a heuristic would be unlikely to end up at one of these apparently inaccessible nodes; alternatively, it could be that the global optima are well connected nodes (or even network *hubs*), but their network neighbours exist in lower-quality fitness levels.

Casting our attention to the LON in Figure 1b —– produced by *walkSample* —– it is a markedly more dense network. There are significantly more nodes and edges between them. Recalling that this is only the highest-quality 0.05% of local optima which were sampled, the implication is that this sampling method exposes much of the neighbourhood (at the local optima level) surrounding the global optimum, and may be more suited to characterising this promising area of the search space. A global optimum can be seen in the centre Figure 1b.

3.3 Heuristics

To gain a view of empirical complexity of the chosen problem instances, we must collect optimisation data by competent heuristic algorithms.

Two competitive algorithms from the literature are deployed: Stützle's Improved ILS for the QAP (IILS) [9], and Taillard's Robust Taboo Search for the QAP (TS) [10]. IILS comes with a wealth of viable parameter configurations. We choose here first improvement hill-climbing, in combination with a perturbation strength of $\frac{3n}{4}$ pairwise exchanges of facilities.

To quantify their performance on the thirty problems, we use the performance gap metric, which is the obtained fitness as a proportion of the desired fitness after a fixed budget of iterations (1000 here). We take the average of this value over 1000 runs. In the results that follow, this metric is referred to as *IILSp* and *TSp* (shorthand for ILS and TS performance).

3.4 Predictive Model Setup

We use both linear mixed models and random forest regression models for algorithm performance prediction. When considering our full (90 sampled LON per sampling algorithm) sets of observations, we conduct random repeated subsampling cross-validation (also known as bootstrapping) for 1000 iterations, each time shuffling the observations randomly. We do this with an 80-20 training-test split. The predictors are normalised as $p = \frac{(p-E(p))}{sd(p)}$. For the smaller datasets (split by parameter set), we only have 30 observations, so we do not split into training and test data, but use all 30 for regression. We still use bootstrapping to average the model statistics over 1000 iterations.

In the results that follow, we focus on the adjusted R^2 associated with each of the models, which quantifies the amount of variance in the response variable (in our case either *IILSp* or *TSp*) explainable using the set of features in question (either the full metric set, which is the union of the network and funnel features, or the funnel or network features exclusively).

For the random forest regression models, we also report the best four predictors for the models in terms of importance, computed during the model formation. We do not conduct random forest regression on the smaller sets of 30

LONs — only on our two full sets of 90 LONs — to provide enough data for the process.

4 Results

4.1 Sampling Algorithm Comparison

To have two sampling algorithms produce networks which are somehow correlated would be encouraging. We rely on sampling algorithms for local optima networks on problems of any reasonable size. As a result, before the algorithms reach real-world applications, they must be refined. Table 1 provides Spearman correlation coefficients, calculated between features of LONs obtained by *optSample* and *walkSample*, respectively. An indication of the p-value is given, as described in the caption. Both network features and funnel features are included as variables. These were introduced in Sections 2.1 and 2.2, respectively.

Most of the features have a fairly weak correlation, but with a p-value less than the threshold 0.05, giving evidence against the null hypothesis that the sample features are unrelated. The correlation for the out-degree of the samples is moderate, and has an encouragingly small p-value of $<$ 0.001. The two strongest associations are for the mean fitness in the network, and the mean fitness of the funnel bottoms (*sinkfitness*), respectively. These two features seem to show a great deal of agreement, gaining very strong correlations indicating statistical significance.

Table 2 attempts to quantify the predictability of *optSample* and *walkSample*. This is done by looking at the ranges of values for important features of the obtained LONs. The minimum value in the sample set is represented as a proportion of the maximum value found.

Looking first at the column giving the information on *optSample*, we can see that the proportions are mostly much nearer to zero than to one. This means that the smallest values are completely dwarfed by the values of the largest. The deduction from this is that you cannot tell this sampling methodology exactly what which trajectory to follow. This is advantageous in the sense that it is not rigid or artificially contained, but could also have the potential to result in unreliable samples. It is, however, promising when considering the wide range of the average outdegree of the obtained networks; this hints at hub-and-spoke network structure being exposed (where it exists).

In contrast, the column showing the range information for *walkSample* shows much larger proportions between 0.29 and 0.99, telling us that this algorithm produces a fairly predictable and therefore tuneable number of nodes and edges. However, because the range for the outdegree is very small (the smallest outdegree is almost as big as the largest, arising due to the nature of the algorithm), important connections between local optima will likely be missed and so-called network 'hubs' may never be found. These are critical because much of the flow of the network — in our case, prospective heuristic search trajectories — may pass through them or be drawn towards them.

Table 1. Spearman's rank correlation, $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$, for pairs of features from the two sampling algorithms.

Predictor	Corr.
optima	0.239^{*}
edges	0.149
ncoptima	0.249^{*}
outdegree	0.441^{***}
diam	-0.212^{*}
meanfitness	0.990^{***}
sinks	0.112
sinkfitness	0.990^{***}
substrength	0.274^{**}

Table 2. The variance (represented as the minimum value as a proportion of the maximum value) of some key metrics for the sampled LONs.

Predictor	*optSample*	*walkSample*
optima	0.0003	0.2499
edges	0.0003	0.2499
outdegree	0.0508	0.9999
sinks	0.0003	0.2499

4.2 Prediction of Heuristic Competence on the Problems

For our predictions, we computed models based on all 90 sampled LONs (that is, 90 per sampling algorithm), which comprise LONs generated using three parameter sets, each of them used for all 30 instances.

We also wanted to see the predictive power when using LONs generated using a single, fixed set of parameters; therefore, we also considered subsets of the LON datasets, where we split into sets of 30.

For *optSample*, the parameter difference came from the perturbation setting used in the construction of the edges, connecting optima in the sample. The three settings of this were $\frac{n}{4}$, $\frac{n}{2}$ and $\frac{3n}{4}$, and the LON sets corresponding to these are referred to as set 1, set 2, and set 3 (in that order) in the models that follow. The set of all 90 LONs together is set 4.

In the case of *walkSample*, the parameter combinations were of the sampling depth, d and length of walk, l. The sets used were (60, 100), (30, 400), and (30, 100); these are referred to as set 1, set 2, and set 3, respectively. Again, the set of all 90 LONs combined is set 4.

Table 3 reports the adjusted R^2 for several predictive models. Each row gives information on the models in question. The columns represent, in order, the sampling algorithm used to obtain the LON samples (*Sample*); the type of predictive methodology (*Model*, one of random forest, *rf*, or linear mixed model, *lm*); the set of features used as predictors (*Features*); the parameter set used in the construction of the network samples (*Param*); and an indication of whether *ILS* or *TS* performance gap was the response variable. The higher the values for the models, the more explainable the variance in the optimisation metric is using that set of predictors.

Table 3. Adjusted R^2 values for linear and random forest models to explain heuristic performance variation on the combinatorial problems.

Sample	Model	Features	Param.	ILS	TS
optSample	lm	all	set 1	0.733	0.425
optSample	lm	all	set 2	0.727	0.592
optSample	lm	all	set 3	0.757	0.734
optSample	lm	all	set 4	0.673	0.539
optSample	lm	funnel	set 1	0.544	0.144
optSample	lm	funnel	set 2	0.428	0.170
optSample	lm	funnel	set 3	0.487	0.265
optSample	lm	funnel	set 4	0.434	0.167
optSample	lm	network	set 1	0.480	0.399
optSample	lm	network	set 2	0.471	0.424
optSample	lm	network	set 3	0.535	0.595
optSample	lm	network	set 4	0.539	0.366
optSample	rf	all	set 4	0.799	0.543
optSample	rf	funnel	set 4	0.615	0.444
optSample	rf	network	set 4	0.678	0.521
walkSample	lm	all	set 1	0.584	0.299
walkSample	lm	all	set 2	0.723	0.809
walkSample	lm	all	set 3	0.673	0.608
walkSample	lm	all	set 4	0.511	0.328
walkSample	lm	funnel	set 1	0.449	0.289
walkSample	lm	funnel	set 2	0.431	0.276
walkSample	lm	funnel	set 3	0.434	0.393
walkSample	lm	funnel	set 4	0.320	0.328
walkSample	lm	network	set 1	0.355	0.086
walkSample	lm	network	set 2	0.445	0.463
walkSample	lm	network	set 3	0.366	0.136
walkSample	lm	network	set 4	0.262	0.098
walkSample	rf	all	set 4	0.497	0.337
walkSample	rf	funnel	set 4	0.274	0.217
walkSample	rf	network	set 4	0.217	0.001

Focusing on the *optSample* entries, we can see that the general trend is that more variance is explained in the ILS than in the TS (compare *ILS* column with *TS* column). When considering the 90 LONs together, using all the features

Table 4. Predictor rankings for the size random forest models.

Sample	Features	Param	Resp.	1	2	3	4
optSample	funnel	set 4	ILS	sinkfitness	ncoptima	funnel	substrength
optSample	network	set 4	ILS	meanfitness	edges	outdegree	diam
optSample	all	set 4	ILS	meanfitness	**sinkfitness**	ncoptima	edges
walkSample	funnel	set 4	ILS	sinkfitness	ncoptima	funnel	substrength
walkSample	network	set 4	ILS	meanfitness	outdegree	edges	diam
walkSample	all	set 4	ILS	meanfitness	**sinkfitness**	outdegree	substrength
optSample	funnel	set 4	TS	sinkfitness	ncoptima	funnel	substrength
optSample	network	set 4	TS	meanfitness	outdegree	edges	diam
optSample	all	set 4	TS	outdegree	**sinkfitness**	edges	meanfitness
walkSample	funnel	set 4	TS	sinkfitness	ncoptima	funnel	substrength
walkSample	network	set 4	TS	meanfitness	outdegree	edges	diam
walkSample	all	set 4	TS	**sinkfitness**	meanfitness	outdegree	ncoptima

together (*all*) produces the strongest models of the three predictor sets (*all*, *funnel*, and *network*). These are followed by the network predictor set models, with the funnel set coming in lowest. This being said, the R^2 values for the full predictor set are higher than for using the network metrics alone, meaning that including the funnel features provides some extra information on the performance of the ILS and TS.

The poorest fits of the *optSample* linear mixed models come from the prediction of TS response using only funnel variables, with only a maximum of 27% of variance being explained. If we compare the random forest models with the linear mixed models — taking the bottom three *optSample* rows against the set 4 (full set) *optSample* rows — it can be seen that these are stronger than the linear predictions.

The *walkSample* models are summarised in the lower half of Table 3. Let us first consider the values obtained using the full set of 90 sampled LONs (the rows where column *Param.* is stated as *set 4*). Comparing these three with one another, we see that for ILS, the best predictor set is all the metrics together. The funnel set work the next-best, with the network set coming in last for predictive power. For Tabu Search prediction, using the full feature set produces an R^2 value of 0.328. Interestingly, using *exclusively* the funnel set produces the same strength of model. Using only the network features has a very poor result, with only 0.098% of variance in performance gap being explained by these.

Just like with the *optSample* models, the trend over the models here is that more of the variance in ILS is explainable with the predictors than that of the TS. This can be seen by locating the *ILS* column and checking against the neighbouring *TS* column entries.

The random forest models for *walkSample* LONs are generally a bit weaker than their linear counterparts (the bottom three rows compared with the *set 4* rows above). In the random forest section, we can see that for these, using the full set of features produces the highest R^2 values. For ILS, using the funnel or network predictor sets is basically comparable (0.274 and 0.217 respectively), but for TS the network set is basically useless (0.001) while the funnel set has some utility (roughly 22% of variance explained).

We now compare the random forest models for the two sampling methodologies and look at six rows in total (all rows where *Model* is *rf*). What is clear, if going by these alone, is that *optSample* has more predictive power for these problems and algorithms than *walkSample*. This can be seen in the uniformly higher R^2 entries.

Doing the same comparison for the linear models for each of our two sampling algorithms (every row where *Model* is stated as *lm*), it is less of a clear-cut distinction. Granted, the general trend is that *optSample* has higher R^2 values, but there are exceptions, notably in the funnel set predictions for Tabu Search, which are poorer than the equivalent models for the *walkSample*.

Table 4 reports a ranking of features in terms of importance in their predictive random forest models. They are ranked from 1-4. The column labelled '1' shows the most important feature from the model specified in that row. In the cases where the full feature set are used as predictors (the rows where *Features* are stated as 'all'), there were eight variables used, but only the best four are reported in the Table.

We can see from examining the '1' row that almost always, the *meanfitness* (mean local optimum fitness in the network) and *sinkfitness* (mean funnel bottom fitness in the network) are ranked as the most important. *Outdegree* seems to be important for prediction of the response variables too.

Looking to the last column, we can see that *substrength* and *diam* populate most of it, indicating these are of lower importance as predictors.

Correlation Study. Figure 2 is a correlation matrix contrasting observed features (*edges*, *meanfitness*, *outdegree*, *diam*, *substrength*, *sinkfitness*, *funnel*, and *ncoptima*), of the local optima networks sampled by *optSample* with heuristic performance on the underlying combinatorial problem (*IILSp* and *TSp*).

Each entry in the upper-right triangle is a Spearman correlation between the row and column variables. The black text is the overall correlation for all 90 LONs, while the coloured text is the correlations for only LONs from a particular parameter set (red is set 1, green is set 2, and blue is set 3). The diagonal shows density plots for the features, with the lower-left triangle showing pairwise scatterplots.

We are, of course, principally interested in the connections between our heuristic performance metrics, and the features of the sampled LONs. The easiest way to approach this is by locating the *IILSp* and *TSp* rows and looking along them, checking against the intersections for interesting correlations. For *IILSp*, we can see that (if we are taking $p < 0.05$ to be indication of statistical significance), many correlations seem notable: *ncoptima*, *funnel*, *sinkfitness*, *substrength*, *diam*, *outdegree*, *meanfitness*, and *edges*.

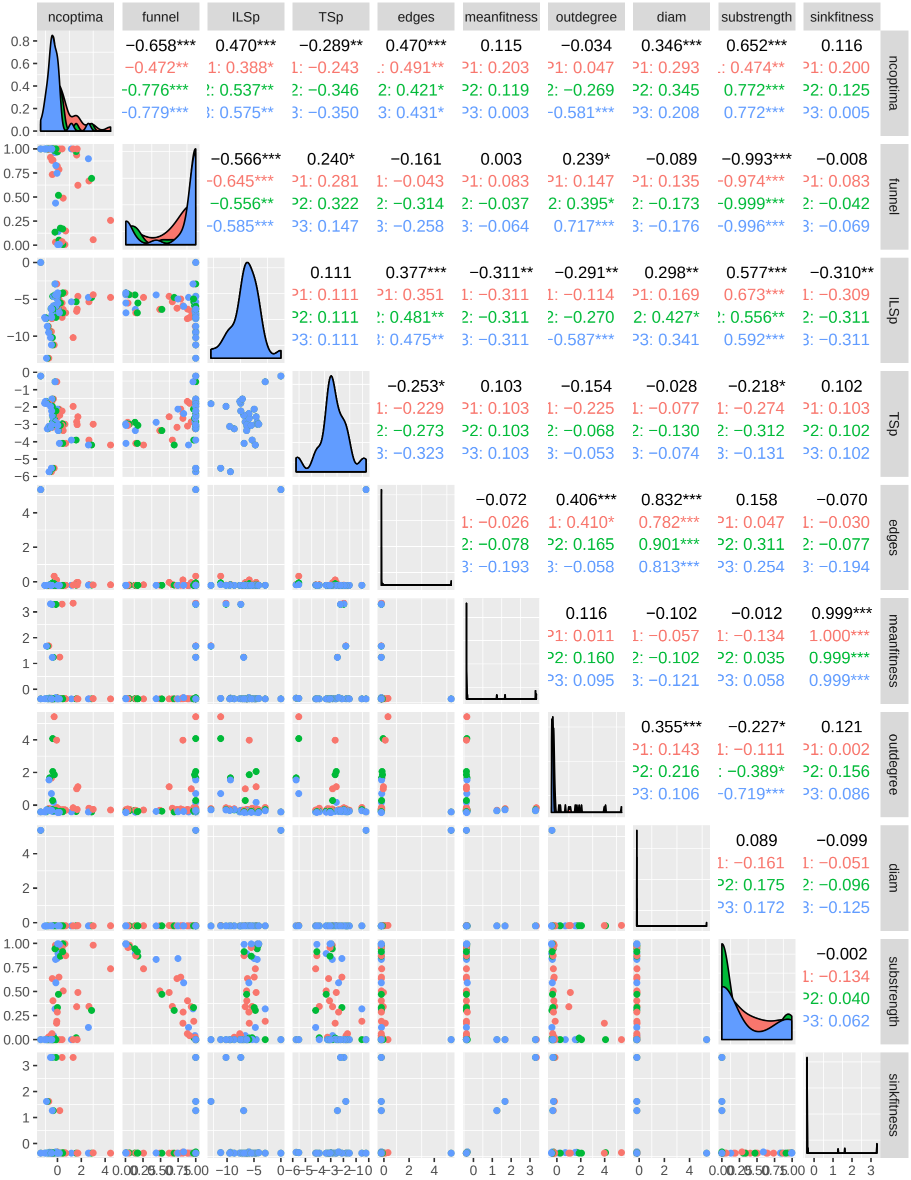


Fig. 2. Correlation matrices of performance metrics and *optSample*-produced LON features. Lower triangle: pairwise scatter plots. Diagonal: density plots. Upper triangle: pairwise Spearman's rank correlation, $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$.

The deduction while reading these is that of those listed, the ones with negative coefficients are correlated to a better heuristic search (a lower performance

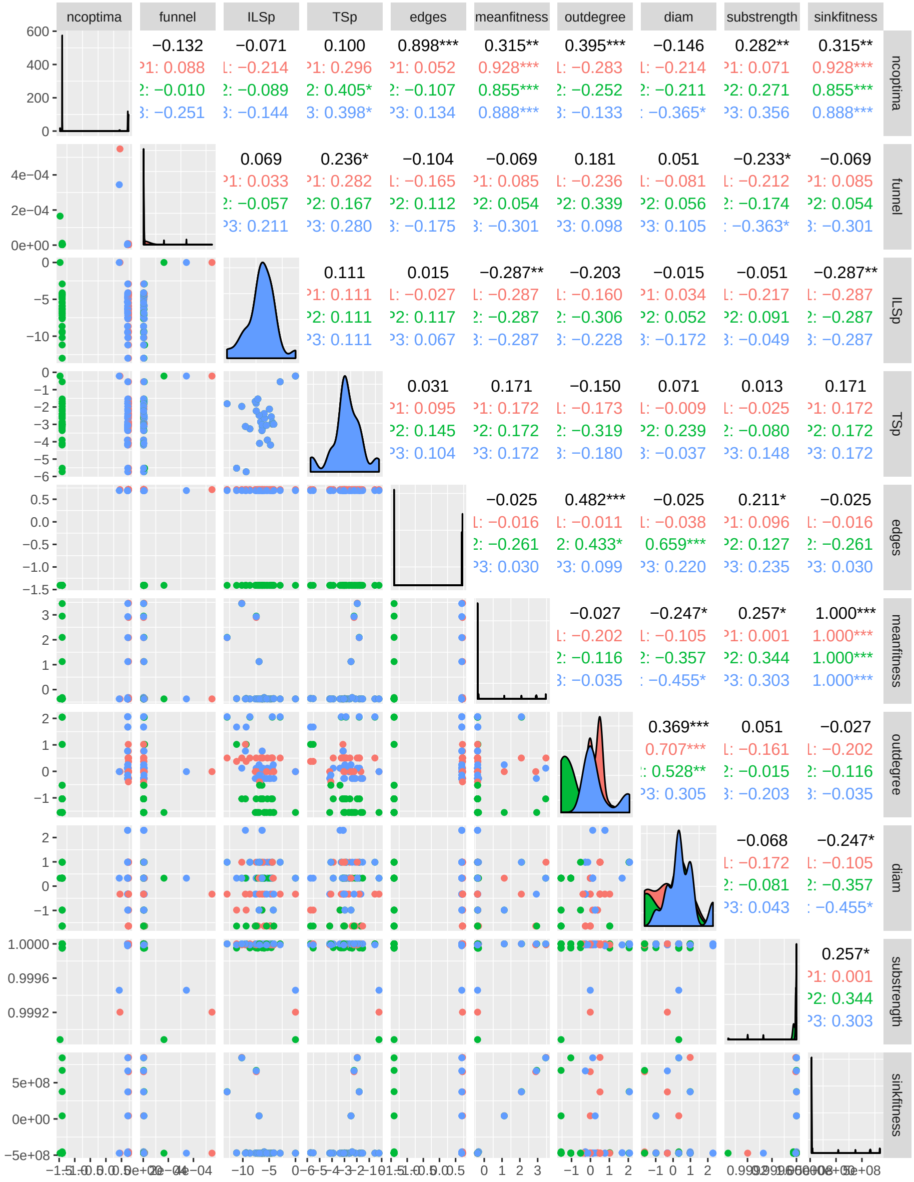


Fig. 3. Correlation matrices of performance metrics and *walkSample*-produced LON features. Lower triangle: pairwise scatter plots. Diagonal: density plots. Upper triangle: pairwise Spearman's rank correlation, $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$.

gap), while those with positive values are correlated with a larger performance gap by our heuristics.

For *IILSp*, those that fall into the first (desirable) category are *funnel*, *sinkfitness*, *outdegree*, and *meanfitness*. Those correlated with a larger performance gap are *ncoptima*, *substrength*, *diam*, and *edges*. In general, the correlations with the features and the performance metrics are stronger in the case of ILS than for TS. However, there are a few to note for TS, selected for their associated p-value: *ncoptima*, *funnel*, *substrength*, and *edges*.

Figure 3 has precisely the same layout as Figure 2, but shows results for *walkSample* instead of *optSample*.

If we assess things in the same way: the intersections of *IILSp* and *TSp* with other columns and rows. In general, there are much weaker correlations with the performance gap data here than we saw for the features of the *optSample* LONs. This can be seen in the smaller values and lack of p-value indications for the feature-performance pairs.

There are a few correlations of note, however: for *IILSp*, there are negative correlations with statistically-sound p-values with *meanfitness* and *sinkfitness* – i.e. if optima, and funnel bottoms, are of a higher-quality fitness, this is correlated with a smaller performance gap attained by the ILS. There is also a positive correlation between *TSp* and *funnel*, indicating funnels are correlated with a larger performance gap for TS.

5 Conclusions and Thoughts

We have conducted an empirical comparison between two sampling algorithms for local optima networks of the Quadratic Assignment Problem. It is essential to refine sampling for local optima networks before we bring them to real world systems. A descriptive statistical comparison was reported, as well as findings pertaining to the algorithms' predictive power in estimating the performance gap obtained by two prominent meta-heuristics on the underlying combinatorial problem.

We found that the two sampling methods exhibited some agreement in the networks they produced and that we could reject the null hypothesis that they produce completely independent samples. They differed from a descriptive perspective in that *walkSample* was tuneable and predictable, while *optSample* varied widely but seemed good at finding hub-and-spoke structure in the local optima space.

A correlation study for the LON features and the heuristic performance metrics was conducted. The correlations were stronger and clear when considering the features of the LONs obtained using *optSample* than *walkSample*. We also worked on predicting heuristic algorithm performance on the problems using linear and random forest models, and found that the sampled LON features (for both *optSample* and *walkSample*) better fit the ILS response variable than the TS one. We saw that generally, including both the *funnel* metric set and the *network* set would be advantageous in explaining search discrepancies for these two heuristics. For both *optSample* and *walkSample*, the extracted funnel metrics proved useful. Going off the random forest models alone, *optSample* uniformly had more predictive power than its competitor, for these choices of instances and heuristics.

From the random forest rankings, the most important predictors were those pertaining to fitness in the sampled networks: the fitness of funnel bottoms, and of nodes in general in the network. This hints that perhaps fitness levels in the local optima space are more pertinent to heuristic search than the subset of transition edges sampled by the LON algorithms.

Acknowledgements. This work is supported by the UK's Engineering and Physical Sciences Research Council (grant number EP/J017515/1). Data generated during this research are available from the Stirling Online Repository for Research Data (http://hdl.handle.net/11667/91).

References

1. Ochoa, G., Tomassini, M., Vérel, S., Darabos, C.: A Study of NK Landscapes' Basins and Local Optima Networks. In: Proceedings of the 10th annual conference on Genetic and evolutionary computation. pp. 555–562. ACM (2008)
2. Ochoa, G., Veerapen, N.: Mapping the Global Structure of TSP Fitness Landscapes. Journal of Heuristics pp. 1–30 (2017)
3. Herrmann, S., Ochoa, G., Rothlauf, F.: Communities of local optima as funnels in fitness landscapes. In: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation. pp. 325–331 (07 2016)
4. McMenemy, P., Veerapen, N., Ochoa, G.: How perturbation strength shapes the global structure of tsp fitness landscapes. In: European Conference on Evolutionary Computation in Combinatorial Optimization. pp. 34–49 (01 2018)
5. Iclanzan, D., Daolio, F., Tomassini, M.: Data-driven local optima network characterization of qaplib instances. In: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation. pp. 453–460. GECCO '14, ACM, New York, NY, USA (2014), http://doi.acm.org/10.1145/2576768.2598275
6. Ochoa, G., Herrmann, S.: Perturbation strength and the global structure of qap fitness landscapes. In: 15th International Conference, Coimbra, Portugal, September 8–12, 2018, Proceedings, Part II. pp. 245–256 (01 2018)
7. Verel, S., Daolio, F., Ochoa, G., Tomassini, M.: Sampling local optima networks of large combinatorial search spaces: The qap case. In: Auger, A., Fonseca, C.M., Lourenço, N., Machado, P., Paquete, L., Whitley, D. (eds.) Parallel Problem Solving from Nature – PPSN XV. pp. 257–268. Springer International Publishing (2018)
8. Stadler, P.F.: Fitness landscapes. In: Biological Evolution and Statistical Physics, pp. 183–204. Springer (2002)
9. Stützle, T.: Iterated local search for the quadratic assignment problem. European Journal of Operational Research 174(3), 1519–1539 (2006)
10. Taillard, É.: Robust taboo search for the quadratic assignment problem. Parallel computing 17(4-5), 443–455 (1991)