On Funnel Depths and Acceptance Criteria in Stochastic Local Search

Sarah L. Thomson
University of Stirling
Stirling, United Kingdom
s.l.thomson@stir.ac.uk

Gabriela Ochoa
University of Stirling
Stirling, United Kingdom
gabriela.ochoa@stir.ac.uk

ABSTRACT

We propose looking at the phenomenon of fitness landscape *funnels* in terms of their depth. In particular, we examine how the depth of funnels in Local Optima Networks (LONs) of benchmark Quadratic Assignment Problem instances relate to metaheuristic performance. Three distinct iterated local search (ILS) acceptance strategies are considered: *better-or-equal* (standard), *annealing-like*, and *restart*. Funnel measurements are analysed for their connection to ILS performance on the underlying combinatorial problems. We communicate the findings through hierarchical clustering of LONs, network visualisations, subgroup analysis, correlation analysis, and Random Forest regression models. The results show that funnel depth is associated with search difficulty, and that there is an interplay between funnel structure and acceptance strategy. Standard and annealing acceptance work better than restart on both deep-funnel and shallow-funnel problems; standard acceptance is the best strategy when optimal funnel(s) are deep, while annealing acceptance is superior when they are shallow. Regression models including funnel depth measurements could explain up to 96% of ILS runtime variance (with annealing-like acceptance). The runtime of ILS with restarts was less explainable using funnel features.

CCS CONCEPTS

• **Mathematics of computing** → **Graph algorithms**; *Combinatorial algorithms*; • **Theory of computation** → **Evolutionary algorithms**.

KEYWORDS

Fitness Landscapes, Quadratic Assignment Problem (QAP), Local Optima Networks (LONs), Funnels, Iterated Local Search, Acceptance Criteria

ACM Reference Format:
Sarah L. Thomson and Gabriela Ochoa. 2022. On Funnel Depths and Acceptance Criteria in Stochastic Local Search. In *Proceedings of Genetic and Evolutionary Computation Conference (GECCO '22).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

GECCO '22, July 9–13, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9237-2 https://doi.org/10.1145/3512290.3528831

1 INTRODUCTION

The study of landscape *funnels* in combinatorial optimisation might be considered a nascent field: how precisely and intimately their geometry relates to search difficulty remains enigmatic.

The notion of a *funnel* was introduced within the protein folding community to describe "a region of configuration space that can be described in terms of a set of downhill pathways that converge on a single low-energy structure or a set of closely-related low-energy structures" [6]. Energy landscapes are conceptually related to fitness landscapes, and funnel structures have also been studied in both continuous optimisation [9, 13, 14] and combinatorial optimisation (see Section 2). The intuition is captured by Figure 1 where two funnels are depicted as two groups of local optima.

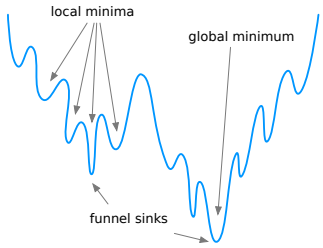


Figure 1: Depiction of two funnels.

A funnel can be defined as a basin of attraction at the level of local optima. One way to analyse these structures is through *Local Optima Networks* (LONs) [18], which are graphs comprising local optima (for nodes) and search transitions between them (for edges). LONs can be constructed using metaheuristic search and therefore reflect landscape dynamics on the associated problem.

A trajectory-based search algorithm which has induced a landscape funnel might become trapped at its terminus, *unless* the algorithm design considers deleterious local optima or restarts. Indeed, the phenomenon of multiple funnel organisations of local optima has been connected to metaheuristic search difficulties [15, 16].

Little is known about the depth of funnels, however. Intuitively, this aspect of their geometry could be critical in understanding their interrelation with algorithm performance. Additionally, we do not yet know the implications of funnel depth on algorithm design. It might be, for example, that particular acceptance strategy approaches are suited to problems with specific funnel depths. In this study, we look to address these nebulous gaps in our knowledge of the nature of funnels.

We consider a well-known benchmark combinatorial optimisation for study: the Quadratic Assignment Problem (QAP). From QAP instances we construct LONs; we also run iterated local search (ILS) variants on them to collect performance information. In order

to study the effect of algorithm design on instances with different funnel depth structure, we consider and compare three separate acceptance strategies within ILS: *better-or-equal* (standard), *annealing*-like, and *restarts*.

From there, the relationship between funnel geometry in the LONs (with particular attention to depth) and ILS performance is examined. Results are presented through visualisation, subgroup analysis, and regression models.

2 FUNNELS IN COMBINATORIAL OPTIMISATION

The notion of funnels in combinatorial optimisation is related to a conjecture proposed in the mid 1990s that the search space of travelling salesman instances had a "globally convex" or "big valley" structure, in which local optima are clustered around one central global optimum [1]. This hypothesis was generally accepted and has inspired the design of some modern search heuristics. The idea of a single valley, however, has been challenged in recent research indicating that the big valley deconstructs into several valleys or funnels [7, 19]. An explicit definition for funnels in combinatorial optimisation using LONs has been proposed [20] and used several times in subsequent works [15–17, 23, 24]; the definition considers a funnel to be an overarching structure leading down to the global optimum.

The occurrence of multiple funnels has been linked to search difficulty: one paper related the number of funnels and the size of the optimal funnel to worsened search [15]; another argued that there was a correlation between search and both the size of the optimal funnel and the flow to its sink (the global optimum) [17]. Despite these advances, to the best of our knowledge, the *depth* of funnels has not yet been studied in this way. Intuitively, this could be of critical importance to search. Furthermore, we also notice that funnel measurements have not been studied through the lens of acceptance strategies in ILS. This paper addresses these vacancies in the literature.

3 DEFINITIONS

3.1 The Quadratic Assignment Problem

A solution to the QAP is generally written as a permutation s of the set $\{1, 2, ..., n\}$, where s_i gives the location of item i. Therefore, the search space is of size $n!$. The cost, or fitness function associated with a permutation s is minimisation and is a quadratic function of the distances between the locations, and the flow between the facilities, $f(s) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{s_i s_j}$, where n denotes the number of facilities/locations and $A = \{a_{ij}\}$ and $B = \{b_{ij}\}$ are the distance and flow matrices, respectively.

We consider all the instances from the QAPLIB[1] [3] with between 25 and 50 facilities; these are of moderate size, and yet are not always trivial to solve. Some of the instances in this group have not been solved to optimality; for those, we use their best-known fitness as the stand-in global optimum. In the rest of this paper, for simplicity we **refer to these as the global optimum**, and to their funnels as global optimal funnels. The QAPLIB instance naming convention has a two digit number *nn* to indicate the problem size. According to [21, 22], most QAPLIB instances can be classified into four types:

(1) *Uniform random distances and flows.* In these problems, denoted by **tainna**, flows and distances are randomly drawn from a uniform distribution.
(2) *Random flows on grids.* These problems consider rectangular distances. The flows are randomly generated, but not necessarily uniformly. These problems are known to be symmetrical and may have multiple different optimal solutions. The instances of this group are: **nug**, **sko**, **tho** and **wil**.
(3) *Real-world problems.* These problems arise from practical applications. The names of the instance sets are **bur**, **chr**, **esc**, **kra**, and **ste**.
(4) *Random real-world like problems.* These instances, denoted by **tainnb**, are randomly generated in a way that they resemble the structure of the real-world instances.

We note here that the there is a set of QAP instances identified as **lip** which are not clear members of any of the four groups above. These instances come from problem generators described in [11], which provide asymmetric instances with known optimal solutions.

3.2 Compressed Monotonic LON Model

Neutrality has been observed at the LON level (i.e. connected sets of optima that share the same fitness value) on several combinatorial problems, including the QAP [17]. Therefore, we use here the coarser LON model proposed in [20], which compresses the local optima that are connected by neutrality into single nodes. The relevant definitions are given below.

Local optima. We assume a search space S with a fitness function $f(S)$ and a neighbourhood function $N(s)$. A local optimum, which in the QAP is a minimum, is a solution l such that $\forall s \in N(l)$, $f(l) \leq f(s)$.

Monotonic perturbation edges. Edges are directed and based on the perturbation operator (k-exchange, $k > 2$). There is an edge from local optimum l_1 to local optimum l_2, if l_2 can be obtained after applying a random perturbation (k-exchange) to l_1 followed by local search, and $f(l_2) \leq f(l_1)$. These edges are called *monotonic* as they record only non-deteriorating transitions between optima. Edges are weighted with estimated frequencies of transition. The set of edges is denoted by E.

Compressed local optima. A compressed local optimum is a set of connected nodes with the same fitness value. Two nodes are connected if there is a monotonic perturbation edge between them. The set of connected optima with the same fitness, denoted by CL, corresponds to the set of nodes in the Compressed Monotonic LON model.

Compressed Monotonic LON. Is the directed graph $CMLON = (CL, CE)$, where nodes are compressed local optima CL, and the edges CE are aggregated from the monotonic edge set E by summing up the edge weights.

Monotonic Sequence. A monotonic sequence is a path of connected nodes $MS = \{cl_1, cl_2, \ldots, cl_s\}$ where $cl_i \in CL$. By definition of the edges, $f(cl_i) \leq f(cl_{i-1})$. There is a natural end to every monotonic sequence, cl_s, when no improving transitions can be

[1] http://www.seas.upenn.edu/qaplib/

found. This node, cl_s, is called a *sink* as it does not have outgoing edges[2].

Funnel. A funnel in the CMLON comprises the aggregation of all monotonic sequences ending at the same point (or sink).

3.3 Funnel Depths and LON metrics

Some of the LON metrics we calculate relate to properties of the optimal funnel. As stipulated in Section 3.1, in certain cases the global optimum is not known and for those instances, the label "optimal" refers to the best-known solution instead. To calculate funnel depth measurements, we first compute all finite shortest paths in the CMLON from *origin* nodes (local optima at the beginning of ILS runs) towards *sink* nodes. A sink is the apex of a funnel. This process is completed separately for *optimal* or *suboptimal* sinks. From the resultant sets of shortest paths (which are monotonic in nature) we can extract metrics. For both optimal and suboptimal funnels, we consider the mean depth and the maximum depth of them, i.e., the mean and maximum of the shortest paths which reach them from origins.

Other LON metrics are included too: the incoming strength to optimal sinks, and the relative size (in nodes) of the optimal funnel(s). The former is calculated as the weighted incoming degree to optimal sinks in CMLON — as a proportion of the total weighted incoming degree to *all* sinks. The size of optimal funnel(s) is defined to be the number of nodes which can reach the optimal sinks using a monotonic path; the size is normalised by the total number of nodes. Also considered is the landscape ruggedness (number of local optima, simply the number of nodes).

4 EXPERIMENTAL SETUP

4.1 Iterated Local Search

We use Stützle's iterated local search (ILS) for both gathering performance data and as the foundation of LON construction [21]. The local search stage uses a strict first improvement hill-climbing variant with the pairwise (2-exchange) neighbourhood. This operator swaps any two positions in a permutation. The perturbation operator exchanges k randomly chosen items. We configure perturbation strength as $k = \frac{N}{2}$, with N being the problem dimension — for both constructing the LONs and computing the performance metrics. This setting was selected as large perturbation strengths of around 50% of N are generally advantageous in this problem domain [17, 21].

4.1.1 Acceptance strategies. Performance metrics were computed using three distinct acceptance strategies for the ILS. This design decision came about because we hypothesised that acceptance strategy might have some sort of relation to funnel deepness.

Better or equal. Only local optima which have improved or equal fitness to the current are accepted. Worsening local optima are never accepted.

Annealing. Improving and equal local optima are always accepted. Worsening local optima are accepted according to a probability governed by an annealing-like cooling schedule. The parameters of the cooling schedule are those which were present in the ILS for QAP algorithm code [21] and are as follows: initial temperature 0.5; end temperature 0.001; number of iterations at a temperature 25; and alpha, 0.8.

Restart. Improving and equal local optima are always accepted. If iterations since an improvement in local optima quality have exceeded $3N$, a restart happens.

4.1.2 Algorithm Performance Metrics. We compute two metrics to summarise ILS performance on the instances. Runs terminate when *either* the known best fitness is found or after 10000 iterations (i.e., hill-climbing followed by perturbation) with no improvement. Because three acceptance strategies are separately employed, there are correspondingly six performance metrics. Each of them is the *mean* over 100 runs starting at random solutions — this is not the same set for each ILS variant. The measurements are: *runtime*, which is the number of iterations upon termination; and *success rate*, a normalised value — the proportion of successful runs (runs where ILS reached the best-known fitness without reaching 10000 iterations with no improvement).

4.2 LON Construction and Metrics

The LON models are constructed by aggregating the unique nodes and edges encountered during 100 independent ILS runs with the standard acceptance strategy. These are distinct from the algorithm performance runs. Runs terminate after 10000 non-improving iterations; this is in order to empirically estimate the end of funnels.

Functions from the R package **igraph** assist in calculating the LON metrics. The shortest paths which are required to calculate funnel depths are computed using **distances** (computes pairs of edge distances between nodes in networks); the function **strength** (sums up the amount of edges are their weight to a given node) facilitates calculation of optimal funnel(s) strength; and **subcomponent** (identifies all nodes which are reachable from the chosen node) determines the optimal funnel(s) size.

At this stage, two instances were removed from the set: **esc32e** and **esc32f**; their local optima networks are uninteresting to study because we found that every node has the same (optimal) fitness. Removing these anomalies left us with the remaining moderate-size (between 25 and 50, inclusive) QAPLIB: 46 instances. Also, some of the **esc** instance LONs contained large global optimum plateaus; consequently, during the computation of their funnel depths, we noticed that certain funnel depths were zero. This is due to some global optima which were both the *origin* of the search (i.e., the first local optimum obtained after improving the initial random solution) and also the termination point, or *sink*. For the purposes of analysis, we remove any length-zero funnel depths before computing the mean and maximum depths for the LONs. This was conducted so that the metrics can intuitively represent the funnel depth *if* there is in fact a funnel present.

4.3 Correlations and Predictive Models

4.3.1 Correlations. For our correlation analysis, a metric which does not hold the assumption of normality must be used: the variables are not normally distributed. This fact can be observed from the density plots in the diagonal panels of Figure 4. We therefore

[2]In directed graphs, a node without outgoing edges is called a *sink*.

use the non-parametric Spearman's rank correlation coefficient [25] and indicate the associated p-values.

4.3.2 Predictive Models. Predictive modelling is conducted with regression using the **randomForest** package [12] in R statistical programming language. Random Forests [2] include design mechanisms intended to prevent overfitting to the training data: *bootstrapping* (re-sampling of the training instances), and sub-setting of the independent variables. These overfitting-prevention mechanisms are the reason Random Forest is chosen for this work. A test set of 20% is kept aside during model training and is then tested on to obtain quality measurements. The training set is selected randomly and without replacement; the test set is the remaining rows. Random sampling is important here, because "similar" LONs (those sharing a name prefix such as **esc**) are next to each other in the dataset.

Candidate independent variables. The features are:

(1) Number of local optima: *local.optima*
(2) Maximum optimal funnel depth: *depth.gfunnel.max*
(3) Mean optimal funnel depth: *depth.gfunnel.avg*
(4) Maximum sub-optimal funnel depth: *depth.sofunnel.max*
(5) Mean sub-optimal funnel depth: *depth.sofunnel.avg*
(6) Incoming strength to optimal sinks: *strength.gfunnel*
(7) Size of optimal funnels, in proportion to the number of nodes: *size.gfunnel*

Iterated local search *runtimes* on the instances serve as response variables, making this a regression setting. Using the other ILS performance variable, success rate, resulted in very poor models; these are therefore not shown. We think that this is because a success rate of 1.0 is a common value. For modelling purposes, the runtimes are taken as their natural logarithm so that resultant error values are more easy to interpret. We aimed for models with as few independent variables as possible, owing to the limited number of eligible QAPLIB instances of moderate size. The *one-in-ten* rule [8] stipulates that roughly ten observations are required per independent variable. Our training set is of size 36 — so we correspondingly set the maximum number of independents as three and conduct feature selection, as described now.

Recursive Feature Elimination. Backwards *recursive feature elimination* (RFE) was used to select model configurations with subsets of the predictors. We employ the R package **caret** [10] for this purpose, and use Root Mean Squared Error (RMSE) as the quality metric for model comparisons. RMSE is the square-root of the MSE, which itself is the mean squared difference between the predicted values and true values. For the experiments, we configure RFE as follows. Random Forest is the modelling method, and only the training data (80%) is supplied. The number of repeats is set at 10; we consider feature subset sizes of one, two, and three from a set of seven candidates (listed earlier). The RFE cross-validation is set to 10-fold; consequently, quality metrics are the mean and standard deviation over 10 validation folds. Two such metrics are reported to accompany the models selected by RFE: RMSE, and the R-Squared (R^2, computed as $1 - \frac{MSE}{variance(t)}$, where t is the response variable). R^2 can be interpreted as the proportion of variance explained.

Models using selected features. After feature selection, Random Forest regression is conducted using the selected features only. There are three separate models, owing to the three variants of ILS. Models are trained using the 80% training data, and then tested on the set-aside 20% testing data. Quality metrics are computed from the predictions made on the test set. The first included measurement is R^2, detailed earlier. Also considered is the RMSE, which is easy to interpret because it follows the same unit range as the response variable.

Details. For all feature selection and subsequent modelling, the default hyperparameters for Random Forest in R are used, namely: 500 trees; minimum size of terminal nodes set to five; a sample size of N (the number of observations); resampling with replacement; features considered per split set to one-third of the number of features. Independent variables are standardised as follows: $p = \frac{(p - E(p))}{sd(p)}$, with p being the predictor in question, E the expected value (mean), and sd the standard deviation.

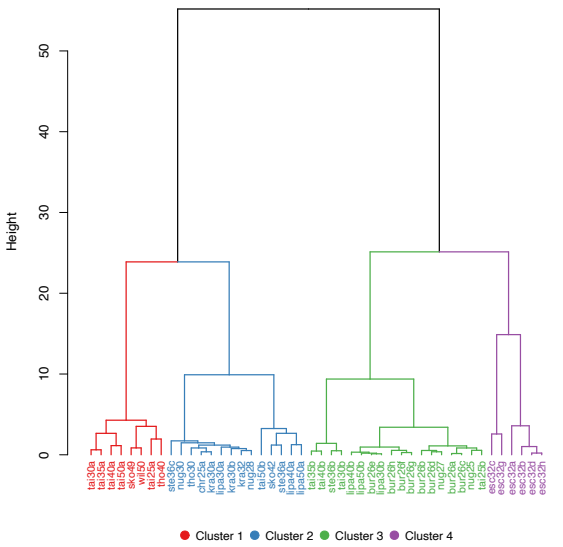
We report model quality measurements rounded to two decimal places — this is for consistency, because the R **randomForest** package returns variance explained (which is part of our results) rounded in this manner.

5 RESULTS

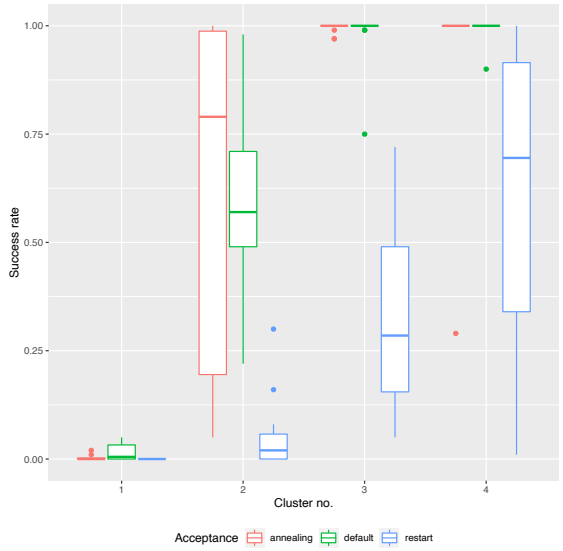
5.1 Clustering Analysis

This section explores whether LON features provide a way of organising the overall (funnel) structure of QAP fitness landscapes into groups, and whether these groups relate to both the types of QAP instances discussed in Section 3.1, and to the ILS performance with the studied acceptance strategies. We use hierarchical clustering to identify groups according to the Euclidean distances between the LON numerical features. We used a wider set of LON features, not only the funnel depth metrics, as the wider set produced a clearer clustering. Specifically, we used the 7 LON metrics described in Section 4.3.2 plus the following 5: number of compressed nodes, mean size of compressed nodes, number of global optima, number of global funnels and number of sub-optimal funnels. The results are presented in Figure 2a. Colour indicates cluster membership and each QAPLIB instance is labelled. We set the number of clusters to $K = 4$ after visually inspecting the dendrogram. Furthermore, Figure 2b shows the distribution of ILS success rate with different acceptance strategies across the clusters. We can see that the restart strategy performs the worst in all cases. The default strategy has a slightly higher success rate on the instances in cluster 1, while for cluster 2 the annealing strategy has superior performance.

The four clusters contain, respectively: 8 (red), 14 (blue), 18 (red) and 6 (purple) instances. A close inspection reveals that cluster 1 (red) contains all the *uniform random distances and flows* instances **tainna**, as well as three of the largest *random flows on grids* instances. Cluster 1 thus contains the hardest instances, as can be confirmed by looking at the very low ILS success rate in this cluster (Fig. 2b). Cluster 2 (blue) groups the rest of the *random flows on grids*, some of the *real-world* instances and the largest **lip** instances. These instances are of intermediate difficulty in terms of success rate (Fig. 2b). Notice that for cluster 2 the annealing acceptance strategy produces higher mean success rate. We argue that this is



(a) Hierarchical clustering



(b) ILS success rates split by cluster and acceptance strategy

Figure 2: Hierarchical clustering of instances using Euclidean distances between standardised LON features. Colours correspond to the order in which results are presented.

due to the presence of suboptimal funnels (see Figures 3b and 3c for visualisations of LONs from this cluster). Accepting worsening solutions, as is the case with annealing acceptance, may help in escaping suboptimal funnels. Cluster 3 contains the rest of the *real-world* instances as well as all the *random real-world like problems* **tainnb**. Finally, Cluster 4 (purple) contains all the **esc** instances, which were found to have high neutrality and very shallow funnels (see Fig. 3f). Thus, clusters 3 and 4 contain the easy instances with very high ILS success rate for both the default and annealing strategies. Collectively, these observations suggest that certain QAPLIB instance types (as described in Section 3.1) lend to similar LONs, indicating that the model captures important landscape characteristics. Moreover, the best choice of the acceptance criterion may be related to the LON (funnel) structure.

5.2 Network Visualisation

Networks are a powerful means of representing patterns of connection, and visualising them can bring useful insight into their structure. Figure 3 illustrates CMLONs for selected QAP instances with different funnel structure. In the images, each node is a compressed optimum, and edges are monotonic perturbation transitions. Plots were produced with R using *force-directed* layout methods as implemented in the igraph library [4]. The decorations reflect features relevant to search — the size of nodes is proportional to their incoming weighted degree (strength), which indicates how much a node 'attracts' the search process. Red nodes belong to the global optimal funnel(s), while blue nodes belong to suboptimal funnels. In the case of **tai30a** (Figure 3a) and **sko42** (Figure 3b), the global optimum is not known and the best-known fitness is used instead. The funnels' terminating nodes (sinks) are highlighted with a more intense colour. The plots in Fig. 3 are organised from left to right according to increasing ILS success rate with the default strategy. The top row shows hard instances (in clusters 1 and 2), while the bottom row shows easy instances (in clusters 3 and 4). The hard instances have a large number of suboptimal funnels (visualised in blue) and longer global funnel depths, while the easy instances lack suboptimal funnels and have shorter global funnel depths.

5.3 Correlation Study

Figure 4 shows correlations for pairs of variables: performance metrics and LON features. Abbreviations used for the LON metrics are detailed in Section 4.3.2. Metrics for ILS runtime on the instances (specifics in Section 4.1.2) begin with **ILS**. There are three of them — one for each acceptance strategy, as described in Section 4.1.1.

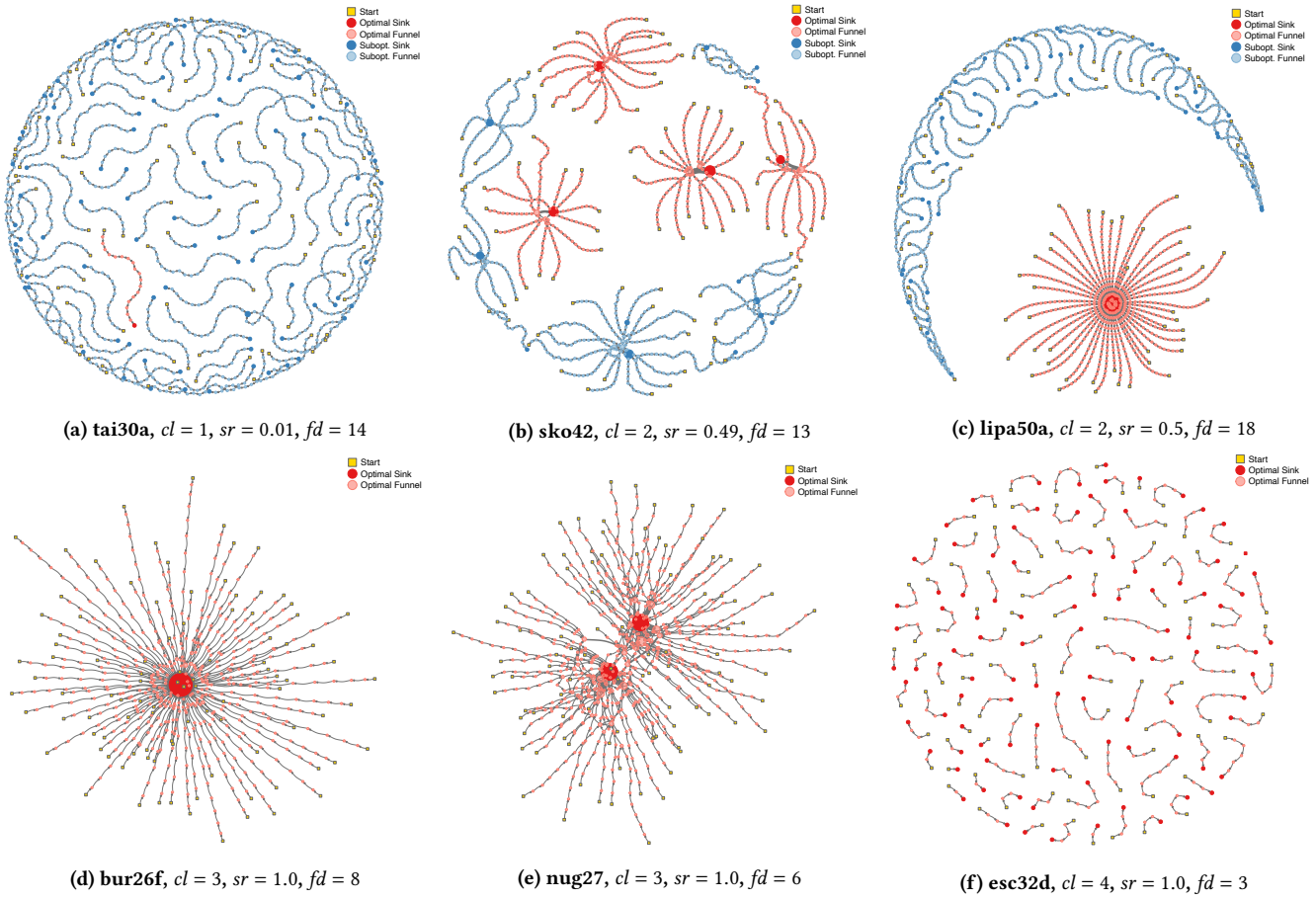
ILS.default.iters is "better-or-equal" acceptance;
ILS.annealing.iters is annealing acceptance;
ILS.restart.iters is a restart strategy.

In the Figure, the lower triangle contains pairwise scatter plots. On the diagonal is density plots, and the upper triangle presents the pairwise Spearman rank correlation, r, with indication of p-value: $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$.

Of particular interest to us is any correlations between algorithm performance and funnel depth measurements. These can be observed by checking the intersections between ILS columns (the last three in the Figure) and funnel metric rows (rows 2-5, labelled on the right).

The correlations show that there are strong, positive associations between ILS runtimes and three funnel depth metrics:

depth.gfunnel.avg, *depth.sofunnel.max*, and *depth.sofunnel.avg*. This implies that these are correlated with longer runtimes, that is, more search difficulty. We also computed correlations between the other ILS performance metric included in this study (success rate, detailed in Section 4.1.2) and funnel depth features. The correlations were similarly strong there (although negative in nature); they are not presented in Figure 4 in the interest of space, but fell



(a) tai30a, $cl = 1$, $sr = 0.01$, $fd = 14$ **(b) sko42,** $cl = 2$, $sr = 0.49$, $fd = 13$ **(c) lipa50a,** $cl = 2$, $sr = 0.5$, $fd = 18$

(d) bur26f, $cl = 3$, $sr = 1.0$, $fd = 8$ **(e) nug27,** $cl = 3$, $sr = 1.0$, $fd = 6$ **(f) esc32d,** $cl = 4$, $sr = 1.0$, $fd = 3$

Figure 3: Local optima network visualisations of selected QAP instances with different funnel structure. For each instance, the cluster membership (*cl*), the ILS success rate with the default strategy (*sr*), and the mean optimal funnel depth (rounded to the nearest integer *fd*) are indicated.

between −0.946 and −0.787 for the three funnel depth measurements just mentioned. Likewise we calculated correlations relating to *strength.gfunnel* and *size.gfunnel* — listed in Section 4.3.2 — and noted that these display strong negative correlations with runtime: between −0.926 and −0.849, and strong positive correlations with success rate: between 0.833 and 0.960.

Notice in the Figure that when compared to the other ILS variables, the correlations are stronger between *ILS.default.iters* and sub-optimal funnel depth measurements (*depth.sofunnel.avg* and *depth.sofunnel.max*). That ILS variant used a "better-or-equal" acceptance strategy. This suggests that this algorithm design is highly related to suboptimal funnel geometry. Every correlation discussed has associated $^{***}p < 0.001$. Operating under the assumption that 0.05 is a reasonable maximum limit for statistical significance [5], we can posit that these correlations appear to be significant for this data sample.

5.4 Funnel Depths and Acceptance Criteria

In this Section, the considered QAPLIB instances are dichotomised by funnel depth for subgroup analysis. Optimal funnels — instead of suboptimal — serve as the factors for this partitioning because some LONs do not contain any suboptimal funnels. One of the two sets comprises instances whose LONs contain deeper optimal funnel(s): where the mean optimal funnel depth is *greater than* the mean for this variable over the whole set of LONs. The other instance set contains all others, i.e., instances whose LONs contain shallower optimal funnels. After the division, there are two groups consisting of 23 instances each. We intend to study how funnel depth relates to acceptance criteria in ILS — to do so, with each group of 23 instances we compute the mean for performance metrics and present the results in Table 1.

Notice by comparing columns one and three in the Table that, for all three acceptance strategies (rows), ILS required less runtime on the shallower funnel(s) instances group. In the case of better-or-equal acceptance, the average number of iterations required for shallow-funnel instances was around 10% of those required for deeper-funnel instances. For annealing, this is even lower: 8.6%. For the restart design, the percentage is much higher: 71.30%. These findings indicate that ILS with a restart strategy is not overly susceptible to funnel depth, but ILS with standard or annealing-like

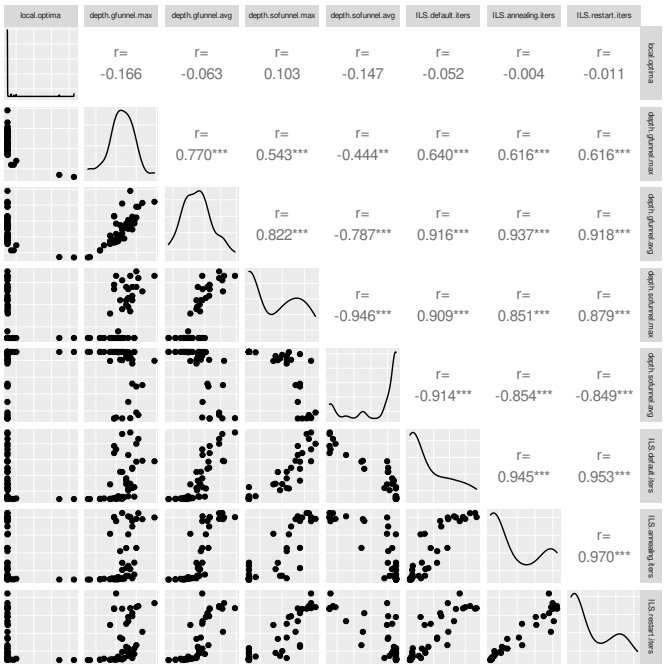


Figure 4: Correlation matrix of ILS runtime performance metrics and LON features.

Table 1: Algorithm performance metrics for ILS on QAPLIB instances, split into two groups: deeper optimal funnel(s) in the LON and shallower optimal funnel(s). Each group contains 23 instances/LONs and the values are the mean in the group. The *runtime* is rounded to the nearest integer (lower values are better), the *success rate* is naturally in the range [0, 1] (larger values are better) Bold text means the best in a column.

	deeper funnel(s)		shallower funnel(s)	
	runtime	*success rate*	*runtime*	*success rate*
better-or-equal	**10200**	**0.4382**	1024	0.9613
annealing	11544	0.4300	**998**	**0.9639**
restart	14723	0.0265	10497	0.4239

acceptance may well be: those ILS variants suffered much longer runtimes on deeper-funnel instances.

For all three acceptance strategies, the success rate is higher for the shallower funnels group (compare column two with column four in Table 1). On instances with LONs containing deeper funnels, better-or-equal and annealing strategies result in success rates 16.5 times and 16.2 times higher (respectively) than using restart strategy (this can be seen by contrasting the entries in column two). In the shallow funnels group, success rates for better-or-equal and annealing are each approximately 2.3 times higher than the rate obtained a restart strategy (column four).

These findings suggest that better-or-equal and annealing-like approaches to acceptance work better than restart on both deep-funnel and shallow-funnel problems; this being said, the difference in performance is orders of magnitude larger when the funnels are deeper. Notice from the bold text (which indicates the best value in a column) that for both runtime and success rate, annealing is the best-performing strategy when the optimal funnel(s) are shallow (columns three and four). When the funnel(s) are deeper, however, better-or-equal acceptance performs best (columns one and two).

5.5 Predictive Models

5.5.1 Feature Selection. Table 2 communicates the feature(s) which were selected by recursive feature elimination (RFE) and quality metrics for the three model setups — each row presents information for a particular setup. The first column indicates the response variable, and the second contains the RMSE for the model setup with its standard deviation. The third column is RMSE taken as a percentage of the range for the response variable. After that is the R-Squared (R^2) with its standard deviation; and finally, the feature(s) selected by RFE from seven candidates (with the maximum allowed being three). Bold text draws attention to the best values within a column.

Notice from the "selected" column of the Table that in all three cases, *mean optimal funnel depth* was among the selected features. This measurement is the average monotonic path length within optimal funnels in the LON. Recall that funnels are comprised of a constellation of monotonic pathways which terminate at the same local optimum *sink* (definition in Section 3.2) — in this case a global optimum.

Table 2: Information about models selected with recursive feature elimination in a Random Forest setting. Provided are model quality metrics on validation data (*RMSE* and R^2) alongside their standard deviations over 10 folds in parentheses. The RMSE%range column also contains the RMSE as a percentage of the response variable range in parentheses. The final column presents the feature(s) which were selected from seven candidates (with the maximum allowed being three). The independent variable candidates are funnel measurements; response variables are the natural logarithm of ILS runtimes.

response variable	RMSE (sd)	RMSE%range	R^2 (sd)	selected
runtime — better-or-equal	1.11 (0.66)	11.2%	**0.86 (0.17)**	[*mean optimal funnel depth maximum suboptimal funnel depth mean suboptimal funnel depth*]
runtime — annealing	1.07 (0.45)	**10.9%**	0.85 (0.19)	[*mean optimal funnel depth*]
runtime — restart	0.09 (0.06)	13.7%	0.84 (0.24)	[*mean optimal funnel depth, maximum suboptimal funnel depth, strength optimal funnel*]

Table 3: Information about predictive models for algorithm runtime estimation using Random Forest regression; model quality metrics are computed on a set-aside test set. The format of the Table mirrors Table 2.

response variable	RMSE	RMSE%range	R^2	predictors
runtime — better-or-equal	0.61	6.2%	0.90	[*mean optimal funnel depth maximum suboptimal funnel depth mean suboptimal funnel depth*]
runtime — annealing	0.41	**4.2%**	**0.96**	[*mean optimal funnel depth*]
runtime — restart	0.10	15.2%	0.73	[*mean optimal funnel depth, maximum suboptimal funnel depth , strength optimal funnel*]

The appearance of this measurement in all three models communicates its salience as a feature to train models for ILS runtime prediction. Another funnel depth feature, *maximum suboptimal funnel depth*, is selected for two models: "runtime — better-or-equal" and "runtime — restart".

Comparing the three rows of the Table, we notice that the strongest models are the first and second, which have ILS runtime with "better-or-equal" and "annealing" acceptance, respectively, as the response variables. These model setups result in the highest R^2 (in the case of the better-or-equal model) and the lowest RMSE (the annealing model). The R^2 for these two convey that around 85-86% of ILS runtime variance is explained with these model configurations.

5.5.2 Models using selected features. Now we build models constructed according to the feature selection conducted in the previous stage. Three predictive models are built using the training data, and then predictions are made on the set-aside test data. Table 3 presents quality metrics computed from these ILS runtime performance predictions.

Two of the three models appear to be of excellent quality: "better-or-equal" ILS runtime prediction, and "annealing" ILS runtime prediction — these have high R^2 values and low relative RMSE (check the R^2 and RMSE%range columns). The strongest model uses the "annealing" ILS design as the target variable, and explains around 96% of variance on the test data.

The weakest model, by a large margin, attempts to predict ILS runtime with the restart strategy (row three). The reason for larger RMSE and lower variance explained is probably that funnel depth metrics cannot explain the length of ILS search when restarts are involved. We posit that this is because restarts regularly "jump" out of funnels by restarting from a new solution. It would follow that the depth of funnels does not matter as much to the search success, at least when compared to ILS with "better-or-equal" or "annealing" acceptance designs.

6 CONCLUSIONS

We considered the interplay between landscape *funnel* depth and metaheuristic algorithm proficiency. The domain was the Quadratic Assignment Problem (QAP) and the metaheuristic was Iterated Local Search (ILS) with different acceptance criteria.

The results showed that funnel depth measurements are related to worsened ILS performance. Correlation analysis and regression models captured this. Regression models using funnel depth properties could explain up to around 96% of ILS runtime variance (using annealing-like acceptance). It appears that ILS with restarts is less affiliated with optimal funnel depth than standard or annealing-like ILS. Annealing acceptance was the best strategy when the optimal funnel(s) were shallow, and standard acceptance was the winner when the optimal funnel(s) were deeper.

As a final note, we acknowledge that larger sizes of QAP instances, and other problem domains, should be studied next. This is necessary in order to provide more evidence that the findings are generalisable. That being said, we are confident that owing to the range of problem sizes and diversity of instances considered in this study, the resultant conclusions apply to QAP instances of moderate size.

Data Publishing. The data from this work is publicly available[3].

[3] https://github.com/sarahlouisethomson/funnel-depths-acceptance-criteria

REFERENCES

[1] K. D. Boese, A. B. Kahng, and S. Muddu. 1994. A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters* 16, 2 (1994), 101–113. https://doi.org/10.1016/0167-6377(94)90065-5

[2] Leo Breiman. 2001. Random forests. *Machine Learning* 45, 1 (2001), 5–32. https://doi.org/10.1023/A:1010933404324

[3] Rainer E. Burkard, Stefan E. Karisch, and Franz Rendl. 1997. QAPLIB – A Quadratic Assignment Problem Library. *Journal of Global Optimization* 10, 4 (1997), 391–403.

[4] G. Csardi and T. Nepusz. 2006. The igraph software package for complex network research. *InterJournal* Complex Systems (2006), 1695.

[5] Giovanni Di Leo and Francesco Sardanelli. 2020. Statistical significance: p value, 0.05 threshold, and applications to radiomics—reasons for a conservative approach. *European radiology experimental* 4, 1 (2020), 1–8.

[6] J P K Doye, M A Miller, and D J Wales. 1999. The double-funnel energy landscape of the 38-atom Lennard-Jones cluster. *Journal of Chemical Physics* 110, 14 (1999), 6896–6906.

[7] D R Hains, L D Whitley, and A E Howe. 2011. Revisiting the big valley search space structure in the TSP. *Journal of the Operational Research Society* 62, 2 (2011), 305–312.

[8] Frank E Harrell Jr, Kerry L Lee, Robert M Califf, David B Pryor, and Robert A Rosati. 1984. Regression modelling strategies for improved prognostic prediction. *Statistics in medicine* 3, 2 (1984), 143–152.

[9] Pascal Kerschke, Mike Preuss, Simon Wessing, and Heike Trautmann. 2015. Detecting Funnel Structures by Means of Exploratory Landscape Analysis. In *Genetic and Evolutionary Computation Conferences* (Madrid, Spain) *(GECCO '15)*. ACM, New York, NY, USA, 265–272.

[10] Max Kuhn. 2008. Building Predictive Models in R Using the caret Package. *Journal of Statistical Software, Articles* 28, 5 (2008), 1–26. https://doi.org/10.18637/jss.v028.i05

[11] Y Li and P M Pardalos. 1992. Generating quadratic assignment test problems with known optimal permutations. *Computational Optimization and Applications* 1, 2 (1992), 163–184.

[12] Andy Liaw and Matthew Wiener. 2002. Classification and Regression by randomForest. *R News* 2, 3 (2002), 18–22. https://CRAN.R-project.org/doc/Rnews/

[13] Marco Locatelli. 2005. On the multilevel structure of global optimization problems. *Computational Optimization and Applications* 30, 1 (2005), 5–22.

[14] Monte Lunacek, Darrell Whitley, and Andrew M. Sutton. 2008. The Impact of Global Structure on Search. In *Parallel Problem Solving from Nature - PPSN X (LNCS, Vol. 5199)*. Springer Berlin Heidelberg, Berlin, Heidelberg, 498–507.

[15] Paul McMenemy, Nadarajen Veerapen, and Gabriela Ochoa. 2018. How perturbation strength shapes the global structure of tsp fitness landscapes. In *European Conference on Evolutionary Computation in Combinatorial Optimization*. Springer, 34–49.

[16] Werner Mostert, Katherine M Malan, Gabriela Ochoa, and Andries P Engelbrecht. 2019. Insights into the feature selection problem using local optima networks. In *European Conference on Evolutionary Computation in Combinatorial Optimization (Part of EvoStar)*. Springer, 147–162.

[17] Gabriela Ochoa and Sebastian Herrmann. 2018. Perturbation strength and the global structure of QAP fitness landscapes. In *International Conference on Parallel Problem Solving from Nature*. Springer, 245–256.

[18] G. Ochoa, M. Tomassini, S. Verel, and C. Darabos. 2008. A Study of NK Landscapes' Basins and Local Optima Networks. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation* (Atlanta, GA, USA) *(GECCO '08)*. ACM, New York, NY, USA, 555–562. https://doi.org/10.1145/1389095.1389204

[19] Gabriela Ochoa and Nadarajen Veerapen. 2016. Deconstructing the Big Valley Search Space Hypothesis. In *Evolutionary Computation in Combinatorial Optimization, EvoCOP 2016 (LNCS, Vol. 9595)*. Springer International Publishing, Cham, 58–73.

[20] Gabriela Ochoa, Nadarajen Veerapen, Fabio Daolio, and Marco Tomassini. 2017. Understanding Phase Transitions with Local Optima Networks: Number Partitioning as a Case Study. In *Evolutionary Computation in Combinatorial Optimization, (EVOCOP) (LNCS, Vol. 10197)*. Springer, 233–248.

[21] Thomas Stützle. 2006. Iterated local search for the quadratic assignment problem. *European Journal of Operational Research* 174, 3 (2006), 1519–1539.

[22] E. Taillard. 1995. Comparison of iterative searches for the quadratic assignment problem. *Location Science* 3, 2 (1995), 87–105.

[23] Sarah L Thomson, Fabio Daolio, and Gabriela Ochoa. 2017. Comparing communities of optima with funnels in combinatorial fitness landscapes. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 377–384.

[24] Nadarajen Veerapen and Gabriela Ochoa. 2018. Visualising the global structure of search landscapes: genetic improvement as a case study. *Genetic programming and evolvable machines* 19, 3 (2018), 317–349.

[25] Jerrold H Zar. 2005. Spearman rank correlation. *Encyclopedia of biostatistics* 7 (2005).