Inferring Future Landscapes:
Sampling the Local Optima Level

Sarah L. Thomson   s.l.thomson@stir.ac.uk
Department of Computing Science and Mathematics, University of Stirling, Stirling,
FK94LA, UK

Gabriela Ochoa   gabriela.ochoa@cs.stir.ac.uk
Department of Computing Science and Mathematics, University of Stirling, Stirling,
FK94LA, UK

Sébastien Verel   verel@univ-littoral.fr
Laboratoire LISIC, Université du Littoral Côte d'Opale, France

Nadarajen Veerapen   nadarajen.veerapen@univ-lille.fr
Univ. Lille, CNRS, Centrale Lille, UMR 9189 - CRIStAL - Centre de Recherche en Informatique Signal et Automatique de Lille, F-59000 Lille, France

Abstract

Connection patterns among *Local Optima Networks* (LONs) can inform heuristic design for optimisation. LON research has predominantly required complete enumeration of a fitness landscape, thereby restricting analysis to problems diminutive in size compared to real-life situations. LON *sampling* algorithms are therefore important. In this paper, we study LON construction algorithms for the Quadratic Assignment Problem (QAP). Using machine learning, we use estimated LON features to predict search performance for competitive heuristics used in the QAP domain. The results show that by using random forest regression, LON construction algorithms produce fitness landscape features which can explain almost all search variance. We find that LON samples better relate to search than enumerated LONs do. The importance of fitness levels of sampled LONs in search predictions is crystallised. Features from LONs produced by different algorithms are combined in predictions for the first time, with promising results for this 'super-sampling': a model to predict tabu search success explained 99% of variance. Arguments are made for the use-case of each LON algorithm and for combining the exploitative process of one with the exploratory optimisation of the other.

Keywords

Combinatorial Optimisation, Fitness Landscapes, Local Optima Networks, Funnel Landscapes.

1 Introduction

A *Local Optima Network* (LON) (Ochoa et al., 2008) is a fitness landscape model used to understand or predict interaction between configuration spaces and search algorithms. LONs consist of local optima for nodes, and the edges are trajectories between local optima. Studying LON objects has brought colour and clarity to our understanding of how optimisation problems and search algorithms interact (Daolio et al., 2011; Ochoa and Veerapen, 2016; Herrmann, 2016; Hernando et al., 2017).

Principally, studies have enumerated the fitness landscape to build a comprehensive LON (Ochoa et al., 2008; Tomassini et al., 2008; Daolio et al., 2011; Verel et al., 2011; Herrmann et al., 2016; Veerapen et al., 2016; Verel et al., 2018). Their focus was on

smaller problems and every candidate solution was mapped to a local optimum within its basin of attraction.

The baseline and the proof-of-concepts were necessary to establish LONs as promising tools. They now have attracted attention (Ochoa and Veerapen, 2018; Bożejko et al., 2018; Herrmann et al., 2018; Fieldsend, 2018; Chicano et al., 2018; Simoncini et al., 2018; Liu et al., 2019), and will likely be applied to increasing numbers of real (and much *larger*) problems. In anticipation of these requirements, refined LON sampling methods are needed. The literature concerning how to sample LONs is in its embryonic stages.

A few LON construction algorithms have been proposed, but have not been extensively tested. We describe here those proposed for the Quadratic Assignment Problem (QAP). Note that in lieu of formal naming by the various authors, we have labelled them as we see fit.

Two-phase LON Sampling. Iclanzan et al. (2014) proposed a LON algorithm to sample n-dimension QAP instances, which had separate phases for recording nodes and edges. Initially, local optima were found by hill-climbing from $2000 \times n$ starting solutions, and thereafter the local optima have a kick or perturbation applied. If the obtained local optimum is also in the node set, an edge is added (or, if it exists already, the edge *weight* is incremented). This algorithm is not used in our study as it seems to be a particular case of the two more recent proposals.

Markov-Chain LON Sampling. Ochoa and Herrmann (2018) introduced a LON algorithm for the QAP built with a competitive heuristic, Iterated Local Search (ILS) (Stützle, 2006). For the remainder of this study, this is labelled *Markov-Chain LON Sampling*, to avoid confusion with the ILS optimisation heuristic itself. The sampling is comprised of multiple ILS runs, each being an adaptive walk on the local optima space. Every local optimum and connection between optima is recorded. The same framework for Travelling Salesman Problem LONs has been used with some success (Ochoa and Veerapen, 2018); this is also instrumented on top of a competitive heuristic in the domain — Chained Lin-Kernighan (Applegate et al., 2003).

Snowball LON Sampling. The same year, Verel et al. (2018) presented a *Snowball LON Sampling* algorithm. A random walk takes place on the local optima level. From each node in the walk, a recursive branching or *snowballing* samples the direct (locally optimal) neighbours. The local optima are found using kicks followed by hill-climbing, as in *Markov-Chain LON Sampling*.

Literature has not asked which method is more representative of the LON being sampled, i.e., are the sampled landscapes similar to those induced by search heuristics during optimisation? There is also vagueness about whether particular LON construction algorithms relate more earnestly to particular search heuristics. A further open issue is that computational cost has not necessarily been considered with regards to LON construction algorithms. Therefore, can their efficiency be compared? And finally, is sampling bias inherent within the methods themselves?

This work has a heavy focus on the relevance of sampled LONs to empirical search difficulty, which manifests as performance prediction using LON features as predictors. We argue this is fruitful in contrast to the analysis of LON features (without aligning them 3 with search variance) — this seems arbitrary and counter-intuitive to the purpose of LONs. Search variance prediction is important because it helps ascribe algorithm proficiency to particular topological features and produces insight about how

algorithmic operators interact with a problem.

The results and experiments of this study come in four parts. First, the comparison of LON samples with fully-enumerated LONs; next LON construction algorithms are provided a fixed computational budget; then LON methods are given various sets of input parameters; and finally the best features from each LON type are taken together into regression models.

This work significantly extends the experiments, analysis and findings of a preliminary, exploratory study into LON construction algorithms (Thomson et al., 2019). The contributions are as follows. Additional problem instances are added to those from the initial QAP study, quadrupling the instance count and more than doubling the problem dimension; extra parameter set inputs for LON construction algorithms are also considered. The first comparison of sampled versus enumerated LONs, in terms of predictive power for search variance is given. For the first time LON construction algorithms are given equal computational budget and the effect studied. We created a new set of features, which contains features obtained with the two different sampling methods, as opposed to sets of features coming from a single method.

The remainder of the article is structured as follows. Section 2 provides the baseline definitions to keep the work self-contained; our methods are in Section 3. Experimental setup is detailed in Section 4, followed by the results in Section 5 and discussion and conclusions in Sections 6 and 7 respectively.

2 Definitions

2.1 Fitness Landscapes

A fitness landscape is both a mathematical object and a metaphor for understanding optimisation processes. For evolutionary computation, a landscape is the tuple (S, N, f) where S is the set of all potential solutions, $N : S \longrightarrow 2^S$, a function that defines the adjacency between solutions, *i.e.* $N(s)$ is the set of neighbouring solutions of the solution s, and f — a fitness function which assigns a fitness value to each solution as $f : S \longrightarrow \mathbb{R}$ (Stadler, 2002). When we consider that search heuristics typically use multiple operators (the operator corresponds to the fitness landscape component N as a notion of connectivity between solutions), it becomes clear that search algorithms induce empirical fitness landscapes that are much more complex than the basic singular neighbourhood (S, N, f) model. Two different search algorithms moving on the same configuration space can manufacture completely dissimilar topological landscape features.

2.2 Local Optima Networks

A sub-space of the fitness landscape is considered, reducing the solution set exclusively to local optima. The neighbourhood now defines the accessibility between local optima often based on the original neighborhood relation of the fitness landscape. This is a *Local Optima Network* as per Ochoa et al. (2008).

2.2.1 Nodes

The set of vertices, LO, consists of local optima (labelled with an index and corresponding fitness). For a minimisation optimisation problem, we define each such local optimum lo_i to satisfy the condition of superior fitness over all other solutions, x, in its neighbourhood: $\forall x \in N(lo_i) : f(x) \leqslant f(lo_i)$, where $N(lo_i)$ is the neighbourhood of lo_i.

2.2.2 Edges

The network edges, E, are weighted and oriented. For *Markov-Chain Sampling* and *Snowball Sampling* LONs, we have an edge if combining perturbation and hill-climbing can transform the source to the destination. The edge weight is the probability of that transformation occurring. Formally, local optima lo_i and lo_j form the source and destination of an edge iff $w_{ij} > 0$.

Joining the nodes and edges induces a LON, $LON = (LO, E)$, a directed graph with nodes $lo_i \in LO$, and there exists an edge $e_{ij} \in E$, with weight w_{ij}, between two nodes lo_i and lo_j iff $w_{ij} > 0$. Note that, in most cases, $w_{ij} \neq w_{ji}$.

2.3 Funnels

Funnels are documented fitness landscape features in combinatorial optimisation (Hains et al., 2011; Ochoa et al., 2017) but originate in the study of physical energy landscapes (Doye et al., 1999). The precise definition in evolutionary computation is an area of active research but a series of papers (Ochoa et al., 2017; Ochoa and Veerapen, 2018; McMenemy et al., 2018; Thomson et al., 2019) consider them to be a basin of attraction at the local optima level. In Figure 1, we see multiple small basins; overall though, these conform to two much larger basins. The large basins are *funnels*, and they contain many local optima organised in a fitness hierarchy. In this minimisation example, there exists a shallower sub-optimal funnel (on the left) and a deeper, optimal funnel (on the right). In use for this work is the associated definition that a *funnel* is the collection of monotonically-improving (in fitness transformations) paths through local optima which terminate at a single local optimum (a *funnel-floor*). To find the paths, the *funnel-floors* are identified from a LON — simply the nodes with no outgoing improving (destination node of edge has superior fitness to source node) edges. From those, a breadth-first search is conducted on the LON, exposing the set of paths which terminate at that funnel-floor. These paths together (both the nodes and the edges) comprise the funnel.
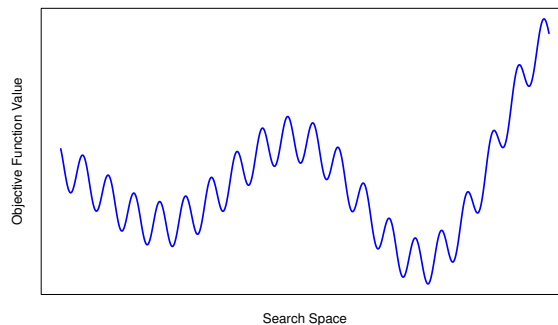

Figure 1: Abstracted from high-dimensional space, a sub-optimal funnel (left) and the *primary* (optimal) funnel on the right, for a minimisation problem.

3 Methodology

3.1 The Quadratic Assignment Problem

A Quadratic Assignment Problem (QAP) (Lawler, 1963) requires allocation of *facilities* to available *locations*. A permutation of facilities assigned to the indexed locations is

the solution representation. Instance specification for a QAP is formed by two matrices: one for pairwise distances between facilities and locations, the other, pairwise flows between them. Mathematically, with the problem dimension being N, the QAP is an assignment of N facilities to N locations where each pair of facilities and locations, $f_i l_j$, have a distance separating them, D_{ij}, and a flow value, F_{ij}. These are encoded in the distance and flow matrices. A solution is a permutation of length N has fitness calculated as the *product* of distances and flows between adjacent assignments. The objective function, g, for a QAP solution x is minimisation of this product where $g(x) = \sum_{i=1}^{N} \sum_{j=1}^{N} D_{ij} F_{ij}, \forall x \in S$. The QAP is NP-Hard and manifests in many real-world problems; it remains a testbed for new fitness landscape analysis (Tayarani-N and Prügel-Bennett, 2016; Verel et al., 2018; Ochoa and Herrmann, 2018). This reality combined with the availability of QAP LON construction algorithms lead us to select QAP for this study.

3.2 Instances

In our analysis we consider the widely-accepted benchmark library, Quadratic Assignment Problem Library (QAPLIB) (Burkard et al., 1997). The library contains 128 instances which also have the optimum fitness evaluations provided. We use 124 of these 128, omitting the largest three due to computational cost (*tho150*, *tai150b*, and *tai256c*), and also *esc16f* because all entries in the distance matrix are zero. The precursor of this work (Thomson et al., 2019) used 30 QAPLIB instances, with a maximum N of 50. Our expanded set results in $N \leqslant 128$. A set of sixty additional instances (N=11) not in QAPLIB also play a part in the results. Their size enables a complete enumeration of the LONs, which is required for comparison to sampled LONs of the same instances. As there is an insufficient number of small instances in the benchmark QAPLIB, this set is desirable to perform statistical analysis.

QAP instances usually conform to four categories (Stützle, 2006), differentiated by distributions in the distance and flow matrices. The four categories are: uniform random distances and flows; random flows on grids; real-world problems; and random real-world like problems. Below we describe our chosen instances in relation to these categories. Note that in QAPLIB, the problem dimension is given in the instance name.

Uniform random distances and flows. Distance and flow values for instances from this category are randomly sampled from a Gaussian distribution. The spread of locations on the plane is random. From QAPLIB we use *tai{12a, 15a, 17a, 20a, 25a, 30a, 35a, 40a, 50a, 60a, 64c, 80a, 100a}* alongside *rou{12, 15, 20}*. Within this class are 30 of our size-11 generated instances.

Random flows on grids. Distance values are organised — locations lie in a defined square on a grid. Flow entries are random. From this category, the QAPLIB instances *had{12,14, 16, 18, 20}*, *nug{12, 14, 16a-b, 17, 18, 20, 21, 22, 24, 30}*, *scr{12, 15, 20}*, *sko{42, 56, 64, 72, 81, 90, 100{a-f}}*, *tho{30,40}*, and *wil{50, 100}* are considered.

Real-world. Instances of the QAP can be formulated from real-world problems. This is the case for some QAPLIB instances. In use in this study are *bur* instances, which comprise of stenotypist typing data; *kra* instances, formulated to plan a hospital layout; *esc* instances, for the testing of sequential circuits; the *ste* set, for fixing units on a grid; and *els19*, which deals with flows of hospital patients moving between departments. Our analysis includes QAPLIB sets *bur{a-h}*, *{els19}*, *esc{16a-e, g-j, 32e, 32g, 128}*, *kra{30a-b, 32}*, *lipa{20a-b, 30a-b, 40a-b, 50a-b, 60a-b, 70a-b, 80a-b, 90a-b}*, and

ste36{a-c}.

Random real-world like. Although not strictly "real-world", these simulate real-world conditions by placing locations in clusters on the plane. In this way, distance values are either small (intra-cluster) or large (inter-cluster). Flows are random. From QAPLIB we use *tai{12b, 15b, 20b, 25b, 30b, 35b, 40b, 50b, 60b, 80b, 100b}*. As above, within this class are 30 of our size-11 generated instances.

Miscellaneous. The QAPLIB *chr* set do not seem to fit neatly into a category. The only constraint is that flows form a mathematical tree. The instances *chr{12a-c, 15a-c, 18a-b, 20a-c, 22a-b, 25a}* are used.

3.3 LON Algorithm 0: Exhaustive Enumeration

The method for exhaustively enumerating a local optima network was introduced alongside the model itself (Ochoa et al., 2008) and then adapted for the QAP (Daolio et al., 2011). LONs are enumerated using a best-improvement algorithm which uses the elementary operator for QAP — a pairwise exchange of facilities. The local optimum x for each solution is found by best-improvement pivot rule and in this way the nodes in the network are added. The escape edges are defined according to the distance d (number of pairwise swaps between solutions), and a threshold for the LON $D > 0$. An edge e_{ij} is traced between x_i and x_j if a solution s exists such that $d(s, x_i) \leq D$ and $h(s) = x_j$. The weight w_{ij} of this edge is $w_{ij} = |\{s \in S : d(s, x_i) \leq D \text{ and } h(s) = x_j\}|$. This weight can be normalised by the number of solutions, $|\{s \in S : d(s, x_i) \leq D\}|$, within reach at distance D. In the present study, we set $D = 2$. The best-improvement algorithm is described below in Algorithm 1.

Algorithm 1 Best-improvement hill-climbing for minimisation problem

---

1: **procedure** HILLCLIMBING
2:     $x \leftarrow$ random initial solution
3:     **while** $x \neq$ Local Optimum **do**
4:         set $x' \in \mathcal{N}(x)$, s.t. $f(x') = min_{y \in \mathcal{N}(x)} f(y)$
5:         **if** $f(x') < f(x)$ **then**
6:             $x \leftarrow x'$
7:         **end if**
8:     **end while**
9: **end procedure**

3.4 LON Algorithm 1: Markov-Chain LON Sampling

We label the first LON sampling candidate (Ochoa and Herrmann, 2018) *Markov-Chain LON Sampling* for the duration of this study. LON nodes and edges are logged during multiple runs of an iterated local search. Each run starts at a random solution, hill-climbs to a local optimum, before 'kicking' or perturbing with a large mutation. Hill-climbing is applied to the perturbed solution to obtain a local optimum, which becomes the input for the next perturbation and hill-climb cycle. The ILS used (Stützle, 2006) is competitive for QAP. To build a LON, 200 runs of ILS track each local optimum and each movement between local optima. Each of these traces is combined to form a single LON.

    The algorithm process is in Algorithm 2. There are parameters which will affect the sample: the number of ILS runs (*runs*), the termination condition (iterations without

improvement, *t*), the pivot rule of hill-climbing (best improvement or first, *hc-type*), and the strength of kick or perturbation (*k*). The sample can be regarded as a joined group of adaptive walks on the local optima level. The multiple start points of the individual runs should — in principle — negate any issue of anisotropy (directional dependency) in the fitness landscape.

Algorithm 2 Markov-Chain LON sampling process

 1: Search space S, Fitness function f,
 2: Perturbation strength k, Stopping threshold t
 3: Choose initial random solution $s_0 \in S$
 4: $l \leftarrow \text{LocalSearch}(s_0)$
 5: $i \leftarrow 0$
 6: **repeat**
 7:     $s' \leftarrow \text{Perturbation}(l, k)$
 8:     $l' \leftarrow \text{LocalSearch}(s')$
 9:     **if** $f(l') \leq f(l)$ **then**
10:         $l \leftarrow l'$
11:         $i \leftarrow 0$
12:     **end if**
13:     $i \leftarrow i + 1$
14: **until** i $\geq$ t **return** l

3.5   LON Algorithm 2: Snowball LON Sampling

The second LON algorithm candidate is called *Snowball LON Sampling*. Put concisely, it is a branching random walk at the local optima level. The process of snowballing has been around for decades in social science (Goodman, 1961) but was not considered for LON sampling until last 2018 (Verel et al., 2018). *Snowball LON Sampling* starts at a random solution and hill-climbs to a local optimum (LO) which is taken to be the first node in a random walk of LO. From this node, a recursive expansion of LO neighbours begins. Specifically, perturbations followed by hill-climbs are taken from the node, to find LO neighbours. All are added to the growing sample. The process is then repeated for the neighbours, to find their own neighbours. Thereafter, *Snowball LON Sampling* returns to the first node on the main walk path, and goes to a random LO neighbour (again by perturbation and hill-climbing) which becomes node two on the walk. The expansion of node two begins. This continues until the random walk is complete.

The pseudocode is shown in Algorithm 3. Parameters exist which can be used to tune the sample: the length of the walk (*l*), the number of neighbours included in a LO expansion (*m*), and the depth of the expansion (number of steps away from original path, *d*).

3.6   LON Features

LON features are chosen with care for our study — a critical component of our analysis utilises them as predictors in regression for performance prediction.

Standard features taken from *Markov-Chain LON Sampling* LONs are the number of LON nodes sampled (denoted as *number optima*); edges sampled (*edges*); mean sampled fitness (*mean fitness*); mean out-degree (mean outgoing edges from nodes, *out-degree*); and the diameter of the LON (longest path between nodes), *diameter*.

Previous work indicated *funnel*-based *Markov-Chain Sampling* LON features were valuable in algorithm performance prediction (Thomson et al., 2019); these are included. Our choices are the proportion of LON edges pointing at the global optimum

---

Algorithm 3 Snowball LON sampling process

1: **procedure** SNOWBALLSAMPLING(d, m, l)
2:    $x_0 \leftarrow hc(x)$ ▷ where x is randomly initialised
3:    $\hat{N} \leftarrow \{x_0\}$
4:    $\hat{E} \leftarrow \emptyset$
5:    **for** $t \leftarrow 0, \ldots l - 1$ **do**
6:        Snowball(d, m, x_t)
7:        $x_{t+1} \leftarrow$ RandomWalkStep(x_t)
8:    **end for**
9: **end procedure**

1: **procedure** SNOWBALL(d, m, x)
2:    **if** $d > 0$ **then**
3:        **for** $j \leftarrow 1, \ldots m$ **do**
4:            $x' \leftarrow hc(op(x))$
5:            $\hat{N} \leftarrow \hat{N} \cup \{x'\}$ ▷ Add node to the sample
6:            **if** $(x, x') \in \hat{E}$ **then**
7:                $\hat{w}_{x,x'} \leftarrow \hat{w}_{x,x'} + 1$
8:            **else**
9:                $\hat{E} \leftarrow \hat{E} \cup \{(x, x')\}$ ▷ Add edge to the sample
10:                $\hat{w}_{x,x'} \leftarrow 1$
11:                Snowball($d - 1$, m, x')
12:            **end if**
13:        **end for**
14:    **end if**
15: **end procedure**

1: **procedure** RANDOMWALKSTEP(x_t)
2:    neighbourSet $\leftarrow \{x : (x_t, x) \in \hat{E} \wedge x \notin \{x_0, \ldots, x_t\}\}$
3:    **if** neighbourSet $\neq \emptyset$ **then**   ▷ Randomly select a neighbour
4:        Select randomly $x_{t+1} \in$ neighbourSet
5:    **else**   ▷ Restart from a random solution x
6:        $x_{t+1} \leftarrow hc(x)$
7:        $\hat{N} \leftarrow \hat{N} \cup \{x_{t+1}\}$
8:    **end if**
9:    **return** x_{t+1}
10: **end procedure**

---

funnel-floor (*incoming global*); and mean funnel-floor fitness (*funnel-floor fitness*).

All features mentioned thus far are useful for *Markov-Chain LON Sampling* LONs but not *Snowball* LONs. *Snowball* samples do not fit well with funnel metrics as heuristic trajectories are uniformly restricted by the nature of the sampling. In fact, the sampling would induce a consistent and increasingly large number of apparent 'funnels' (according to our definition in Section 2.3). The short branching paths during node expansion would also lead to numerous nodes with no outgoing edges, which are technically defined as funnel-floors. In reality, improving transformations from these nodes is likely possible. Similarly, standard features of the samples such as quantities of nodes, edges, and out-degrees in *Snowball Sampling* LONs are redundant as predictors — they are artefacts of the sampling parameters (length of the random walk, number of edges, and depth of snowball expansion).

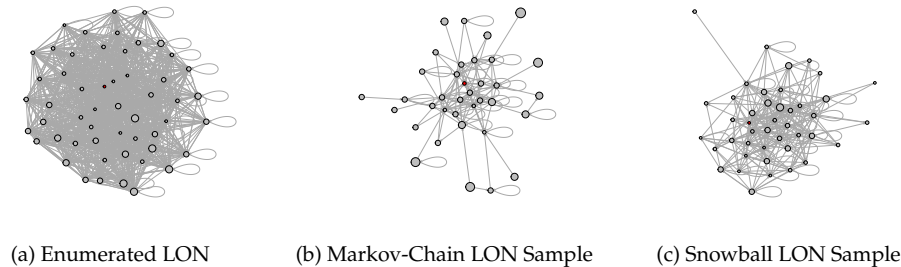We see in Figure 2 that even for a diminutive N = 11 instance, the LON construction

(a) Enumerated LON   (b) Markov-Chain LON Sample   (c) Snowball LON Sample

Figure 2: Three LONs extracted from the same N = 11 QAP instance (from class 'random real-like'). All produced with different algorithms as indicated in the sub-captions. Node size is proportional to fitness (larger is fitter). Global optimum is shown in red, all other nodes in grey.

algorithms capture rather different things. Taking a bird's eye view, the *Markov-Chain* LON seems to be sparser in terms of nodes, edges, and edge density. In fact, the enumerated LON has 53 nodes, 1134 edges; *Markov-Chain LON Sample* has 36 nodes, 122 edges; and *Snowball LON Sample* has 43 nodes, and 272 edges.

For *Snowball LON Sampling*, features are based on the attributes encoded in the nodes and edges: the fitness distribution and the edge weight distribution. These are more appropriate to what the sample has to offer. Metrics based on the density and connection pattern of edges will not give meaningful information because these are an artefact of the chosen sampling parameters, instead of an inherent structure. In particular, funnel metrics calculated on *Snowball* LONs demonstrated that almost all nodes were identified as funnel-floors. Instead, included in the chosen metric set is the mean weight of self-loops (*weight loops*); mean weight disparity of outgoing edges (*weight disparity*); and fitness-fitness correlation between neighbours (*fitness correlation*). Statistics collected during snowballing are included too, namely: mean length of hill-climb to local optimum (*mean HC length*); maximum length of hill-climb to local optimum (*maximum HC length*); and maximum number of paths to local optimum (*maximum HC paths*).

4 Experimental Setup

In this Section we detail the setup of the LON construction algorithms, the predictive models used, and the QAP search algorithms.

4.1 Markov-Chain LON Sampling: Details

For the main QAPLIB study, seven parameter configurations are given to the *Markov-Chain LON sampling* algorithm, producing seven *Markov-Chain* LONs for each QAPLIB instance. Parameter choices are amongst those provided and suggested by the author of the base ILS heuristic (Stützle, 2006), using the form {perturbation, type local search}, and they are: $\{\frac{n}{16}$, first$\}$; $\{\frac{n}{16}$, best$\}$; $\{\frac{n}{8}$, first$\}$; $\{\frac{n}{8}$, best$\}$; $\{\frac{n}{4}$, first$\}$; $\{\frac{n}{2}$, first$\}$; $\{\frac{3n}{4}$, first$\}$. For all, 200 runs are started from a random solution, so that diverse regions of the local optima space will be sampled. Each run terminates when there has not been an improvement in 1000 iterations. During preliminary runs, this condition was found to be sufficiently large such that local optima are not prematurely considered to be in-

escapable *funnel-floors* (recall Section 2.3), but low enough that computational cost is reasonable. Concerning the sensitivity of the results to the parameter choices: if the number of runs were lower, the relation from the constructed LONs to optimiser performance would be proportionally weaker. A lower value for the termination condition would produce LONs riddled with *pseudo*-funnel floors, presenting an image of the local optima space which is more complex than the empirical reality. The full process is given in Section 3 and Algorithm 2.

Seven sampling configurations for 124 problem instances gives us 868 LONs for QAPLIB instances as produced by *Markov-Chain LON Sampling*. In addition, *Markov-Chain LON Sampling* LONs for the 60 synthetic size-11 instances and for the fixed-evaluations QAPLIB LONs were constructed using $\{\frac{n}{2}, \text{first}\}$.

4.2 Snowball LON Sampling: Details

Parameter configurations for *Snowball LON Sampling* lie in the form {l, m, d} and are as follows: {100, 20, 2}; {100, 30, 2}; {100, 50, 2}; {100, 60, 2}; {200, 30, 2}; {400, 30, 2}; and {400, 50, 2}. The choices are based on those suggested by the algorithm's author (Verel et al., 2018). To re-iterate, we start once from a random solution, and hill-climb to a local optimum, which is the first in the random walk. The algorithm will terminate when the walk length and node expansion is complete, as in Section 3 and Algorithm 3. Like *Markov-Chain LON Sampling*, there are 868 (seven parameter sets $\times$ 124 instances) for QAPLIB produced by this algorithm. *Snowball LON Sampling* LONs for the synthetic size-11 instances and for the fixed-evaluations QAPLIB LONs were constructed using {100, 60, 2}.

4.3 Predictive Models: Details

Linear and random forest regression are used for algorithm performance prediction. Linear regression models describe the linear effect of predictors on the response variable; random forest is known for capturing non-linear interactions between variables. For each, random repeated sub-sampling cross-validation (also known as bootstrapping) is conducted for 100 iterations, each time shuffling the observations randomly. We do this with an 80-20 training-test split. The predictors x_j are normalised and reduce to standard deviation one as $\frac{(x_j - E(x_j))}{sd(x_j)}$. The random forest regressions use 500 decision trees; the number of features sampled as candidates during splits is $\frac{1}{3}n$, where n is the number of features.

For the bulk of our models (results Sections 5.2, 5.3, 5.4) the adjusted R^2 is a reported statistic. This quantifies the amount of variance in the response variable — which is the obtained fitness (after 1000 iterations) as a proportion of the desired fitness — which is explainable using the set of predictors. Also in use is the *mean-squared error* (*MSE*), a notion of how accurate predictions from the model are. For the random forest regressions, the predictor *importance rankings* are reported. A portion of our study is comparing LON enumerations with LON samples. For this, we have a different environment for modelling. Because the LON must be fully enumerable, the instances are bounded at N = 11 and are synthetic. There are 60 observations (feature vectors for 60 LONs), arising from two QAP classes (random 'real-like' and uniformly-random, respectively). Accounting for effects caused by QAP class is an objective for this set of experiments (the results are in Section 5.1), in particular because of the limited number of observations. These 'random effects' can be controlled for in a hierarchical linear mixed model, also known as a mixed-effects model.

To formalise the hierarchical modelling approach, let us take y_{ik} to be the search

performance (for example, success rate or runtime) observed on instance i from class k (k is real-like or uniform-random). The linear model is then:

$$y_{ik} = \beta_0 + \sum_{j=1}^{p} \beta_j x_{jik} + \alpha_k + \epsilon_{ik}, \qquad \epsilon_{ik} \sim \mathcal{N}(0, \sigma^2)$$

where x_{jik} is the value of predictor j (e.g., number of local optima, number of funnels, etc.) of instance i from class k, β_j is its corresponding fixed effect on search performance, α_k are the random effects conditional on problem class k, which represent random deviations from the common intercept β_0, and finally ϵ_{ik} are the model residuals.

The summary statistics used for this subset of regression models (Section 5.1) are the *conditional R^2*, the *marginal R^2*, and the *Root Mean Squared Error* (*RMSE*) (given as a proportion of the response variable's range). *Conditional* R^2 is the variance explained by the complete model. *Marginal* R^2 is the ratio of variance that is explained by *only* the fixed effects (Nakagawa and Schielzeth, 2013) (negating or controlling for the random effects from QAP instance class). The fixed effects are the LON features. *RMSE* captures the standard deviation of the model predictions (also known as *residuals*) and is therefore useful for estimating the variability of predictions. RMSE is in the same unit range as the response variable. For ease of interpretation, RMSE is taken as a proportion of the total range for the response variable (search performance).

4.4 QAP Search Algorithms

For our regression response variables, search algorithm performance on the QAP instances is used. The search algorithms are two competitive search algorithms for the QAP: Improved Iterated Local Search (*ILS*, Stützle (2006)), and Robust Taboo Search (*TS*, Taillard (1991)). These are run 100 times on every QAP instance and after 1000 iterations the obtained fitness is taken as a proportion of the desired fitness. This is our measure for how difficult the instance was for *ILS* or *TS* and is the response variable in the models seen in results Sections 5.2, 5.3, and 5.4. This metric does not hold up well though for our N=11 instances: they are easy for *ILS* and *TS*, meaning the obtained fitness is frequently the desired fitness. For those experiments (models reported in results Section 5.1) the metric *number of iterations to reach the global optimum* is instead used as our algorithm response variable.

ILS comes with a wealth of viable parameter configurations. The choices for this study are first improvement hill-climbing, in combination with a perturbation strength of $\frac{3n}{4}$ pairwise exchanges of facilities. This perturbation strength is known to be well-performing (Stützle, 2006) and is chosen to generate a sufficiently different solution from the previous local optimum.

5 Results

This Section reports the experimental results, mostly in the form of regression model summaries. The primary aim is analysing the relationship between the LONs and search algorithm performance.

5.1 Approximating the LON Enumeration

5.1.1 Features

Let us begin a brief look at the relationships between features of LONs produced by the construction algorithms. Figure 3 plots the number of local optima in the LON (right), and the number of edges (left). The scatterplot is the feature (nodes or edges) from LON

enumeration (x-axis) and the same feature from *Markov-Chain LON Sampling* (y-axis). The random 'real-like' LONs are red, uniform random in green.

The two plots share similar trends. The uniform random LONs in green suggest a reasonable correlation between the estimation and the true number of nodes, and edges. The numbers of nodes and edges are, as expected, smaller when produced by the sampling algorithm instead of the enumeration. Although less obvious from the plots, the same is true of the random real-like LONs (in red). In fact, taking all 60 LONs together carries an associated 0.926 Pearson correlation between enumerated node count and sampled node count. The edge count correlation stands lower at 0.793. *Snowball LON Sampling* generates LONs with stronger still correlation plots. There is a 0.996 correlation with enumeration in terms of nodes; 0.977 for edges. These positive trends are seen in Figure 4. Let us clarify that extrapolating from these results to larger instances is difficult. Although *Snowball LON Sampling* often extracts a node set with size representative of the enumerated LON set, this might only be the case for very small instances. The algorithm is based on a single start point; therefore, the sample could potentially reflect a low-quality region of local optima.
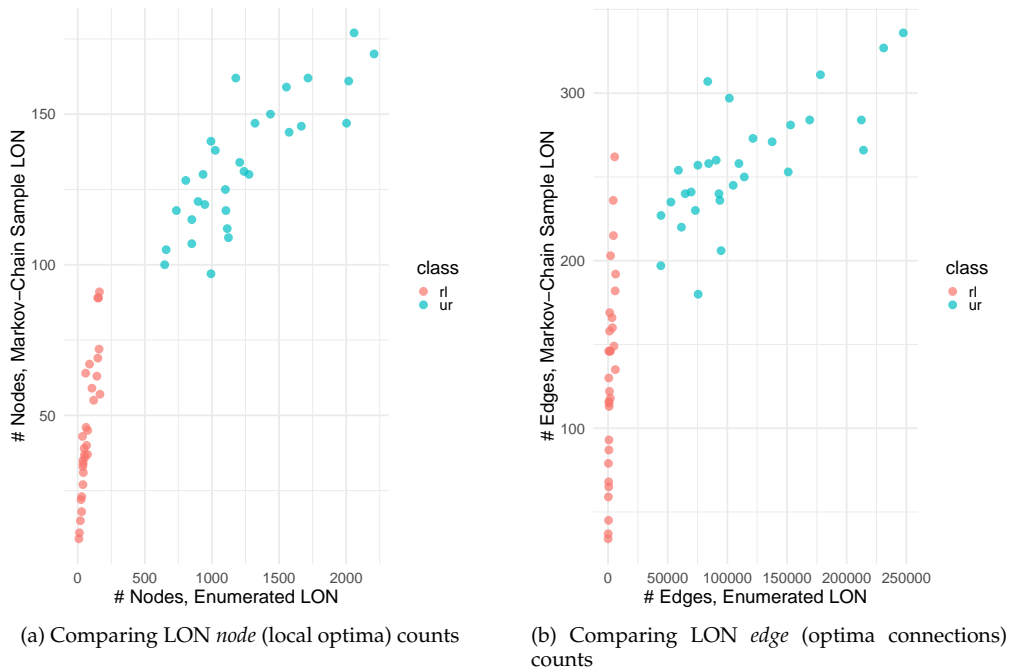

(a) Comparing LON *node* (local optima) counts

(b) Comparing LON *edge* (optima connections) counts

Figure 3: Scatterplots for: features of fully enumerated LONs (x-axis) versus features of sampled *Markov-Chain* LONs (y-axis).

5.1.2 Predictive Models

Consider now Table 1 for performance prediction models using features from different types of LON sampling. In terms of explanation of variance in the $\{ILS, TS\}$ responses (see Marginal R^2 column), the models are somewhat weak after controlling for the random effects (allowing for difference in problem type). This can be seen by comparing the smaller *Marginal* R^2 values with the *Conditional* R^2 ones. The confusion of the response variable is likely due to the small number of observations (60) and smaller still

(a) Comparing LON *node* (local optima) counts

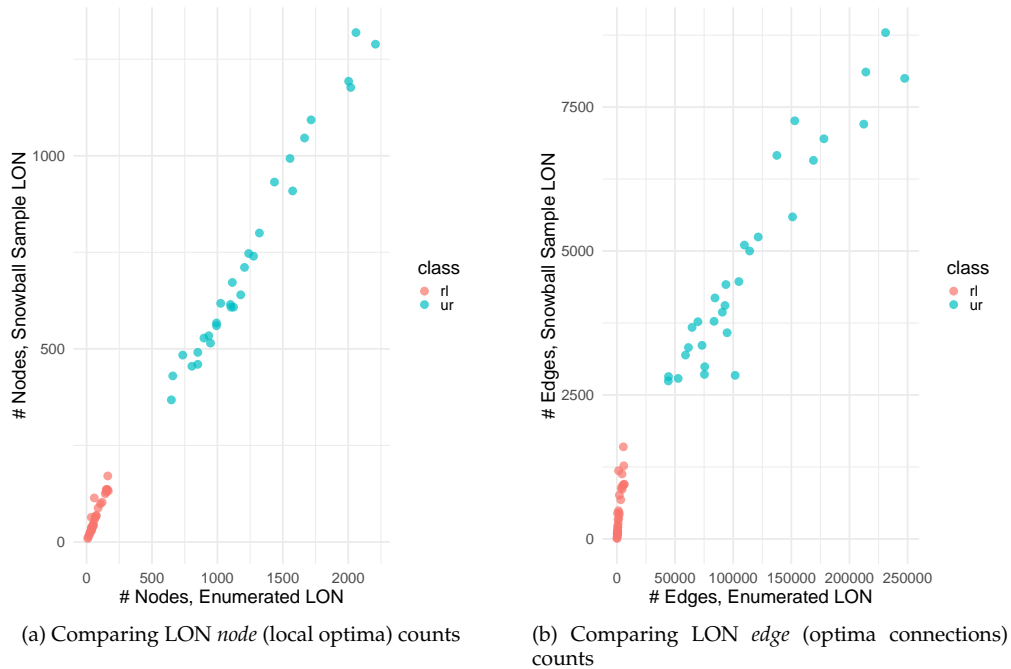(b) Comparing LON *edge* (optima connections) counts

Figure 4: Scatterplots for: features of fully enumerated LONs (x-axis) versus features of sampled *Snowball* LONs (y-axis).

number belonging to a particular class (30 each). The purpose of the models is *comparison* between LON construction algorithms though. The models do have low relative RMSE — typically one fifth of the range of the response variable. Taking the *Marginal R^2* as indication of quality, the two strongest models are using sampled (*Markov-Chain and Snowball*) LONs to predict *ILS* response on the problem. Notice that the *Marginal R^2* and *Conditional R^2* are equal for the former. This means there was no detectable random effects coming from difference in problem class. The result that sampled LONs explain more variance than enumerated is noteable — intuition would tell us that the enumerated LON (rows one and two in the Table) would give superior information about the problem. *Markov-Chain LON Sampling* and *Snowball LON Sampling* are, however, based on operator combinations which are in search algorithms used in practice on the QAP. Enumeration of LONs is simply running a local search from every possible solution to assign a local optimum. This may not mirror actual algorithm-problem interactions. This result is interesting given that most LON work has analysed enumerated LONs (Ochoa et al., 2008; Daolio et al., 2011; Herrmann et al., 2016).

5.2 LON Sampling: Equal Computational Threshold

A previous work (Thomson et al., 2019) compared LON construction algorithms but allowed them unequal computation. Both were given default parameters suggested by authors of the algorithms. For this Section, LONs were produced using a comparable computational budget. Each LON algorithm was allowed 50,000 fitness function evaluations. Now we investigate which provides more predictive information about search difficulties under these restricted conditions.

Table 1: Mixed-effects Model Summary statistics, for the $N = 11$ LONs. Marginal (fixed effects) R^2, conditional (fixed and random effects) R^2, and Root Mean Squared Error are shown.

LON Method	Response Variable	Marginal R^2	Conditional R^2	RMSE
Exhaustive	ILS	0.159	0.832	0.21
Exhaustive	TS	0.148	0.719	0.18
Markov-Chain	ILS	**0.406**	0.406	0.20
Markov-Chain	TS	0.220	0.816	0.20
Snowball	ILS	0.370	0.374	0.20
Snowball	TS	0.147	0.745	0.23

Table 2 gives model summaries, including the LON algorithm used to produce the samples, the type of regression, the chosen response variable (*ILS* or *TS*, meaning the normalised fitness they obtained on the problem), the adjusted R^2, and the mean-squared error (*MSE*). Tables 3 and 4 then record the random forest predictor rankings for *Markov LON Sampling* and *Snowball LON Sampling* respectively. Looking over the

Table 2: Linear and random forest regression model summary statistics, for the fixed-evaluations QAPLIB LONs. Adjusted R_2 and MSE are given.

LON Method	Regression	Response Variable	R^2	MSE
Markov-Chain	Linear	ILS	0.471	0.002
Markov-Chain	Linear	TS	0.336	0.132
Markov-Chain	RandomForest	ILS	0.245	0.002
Markov-Chain	RandomForest	TS	0.154	0.144
Snowball	Linear	ILS	0.303	0.002
Snowball	Linear	TS	0.418	0.024
Snowball	RandomForest	ILS	0.230	0.002
Snowball	RandomForest	TS	**0.521**	0.013

Table 3: Predictor rankings for RF models using fixed-evaluations *Markov-Chain* QAPLIB LONs. Fitness features in italics.

Predicting ILS	Importance Value	Predicting TS	Importance Value
incoming global	0.058	*mean fitness*	2.130
out-degree	0.023	*funnel-floor fitness*	2.127
number optima	0.023	edges	1.742
mean fitness	0.022	number optima	1.736
edges	0.021	out-degree	0.349
funnel-floor fitness	0.019	incoming global	0.304

ILS rows in Table 2, it can be seen that the *MSE* rates are comparable for the LON construction algorithms; however, the R^2 values are higher for both types of regression for *Markov-Chain* over *Snowball*. This being said, *Snowball* LON features predicting *TS* using random forest regression is the strongest model shown, with around 52% of variance explained. This is especially interesting with the consideration that typically the R^2 values are smaller for random forest over linear, but this model is the only exception.

Table 4: Predictor rankings for RF models using fixed-evaluations *Snowball* QAPLIB LONs. Fitness features in italics.

Predicting ILS	Importance Value	Predicting TS	Importance Value
maximum HC length	0.043	maximum HC length	0.656
weight disparity	0.028	*mean fitness*	0.320
mean fitness	0.025	mean HC length	0.310
maximum HC paths	0.017	*fitness correlation*	0.273
fitness correlation	0.015	weight disparity	0.193
weight loops	0.014	weight loops	0.156
mean HC length	0.014	maximum HC paths	0.141

Snowball LON sampling is superior to its competitor when taking tabu search (*TS*) to be the response variable. This can be seen by comparing rows 2 and 4 with rows 6 and 8. The R^2 values are higher and the error rates are smaller (by an order of magnitude) in the *Snowball* models, compared to those of *Markov*.

Now the predictor rankings of Tables 3 and 4 are considered.

5.3   LONs to Predict QAPLIB Problem Difficulty

In this section, the set of 868 LON samplings (for each LON algorithm, totalling 1,736) for QAPLIB is used. Recall that these are associated with 124 QAPLIB instances, each having seven LONs produced per algorithm (differentiated by sampling parameter configuration). Table 5 shows the algorithmic performance prediction model summaries, indicating the strength of the models in terms of adjusted R^2 and *MSE*. Tables 6 and 7 record predictor rankings for the random forest models for *Markov-Chain LON Samples* and *Snowball LON Samples*, respectively. Rows 1 and 5 of Table 5 (linear

Table 5: Linear and random forest regression model summary statistics for full QAPLIB LON set. Adjusted R_2 and MSE are given.

LON Method	Regression	Response Variable	R^2	MSE
Markov-Chain	Linear	ILS	0.043	0.002
Markov-Chain	Linear	TS	0.180	0.081
Markov-Chain	RandomForest	ILS	0.645	0.000
Markov-Chain	RandomForest	TS	**0.925**	0.008
Snowball	Linear	ILS	0.057	0.003
Snowball	Linear	TS	0.252	0.029
Snowball	RandomForest	ILS	0.804	0.000
Snowball	RandomForest	TS	**0.922**	0.003

Table 6: Predictor rankings for main QAPLIB RF models — Markov-Chain LON samples. Fitness features in italics.

Predicting ILS	Predicting TS
mean fitness	*mean fitness*
funnel-floor fitness	*funnel-floor fitness*
number optima	edges
incoming global	diameter
edges	number optima
diameter	out-degree
out-degree	incoming global

Table 7: Predictor rankings for main QAPLIB RF models using *Snowball* LONs

Predicting ILS	Predicting TS
mean fitness	*mean fitness*
maximum HC paths	*fitness correlation*
fitness correlation	maximum HC paths
mean HC length	weight disparity
weight disparity	maximum HC length
maximum HC length	mean HC length
weight loops	weight loops

regression to predict *ILS* response) show that neither *Markov-Chain* nor *Snowball* LON features build a good model. The R^2 values are just too weak. However, when we look at the equivalent random forest models (rows 3 and 7) we have strong models with 76% (*Markov-Chain*) and 80% (*Snowball*) of variance explained, and very small relative *MSE* values. This could reflect the capability of regression trees to capture non-linearity and complex interactions between predictors. The same trend is seen in the tabu search prediction models in Table 5. Linear models (rows 2 and 6) are weak, with small R^2 and comparably larger *MSE*. The random forest results (rows 4 and 8) are strong, with 90% or over variance explained by features of LONs produced by both LON construction algorithms.

Taking *Markov-Chain* and *Snowball* as rivals for predicting *ILS* response, *Snowball* is slightly superior; for *TS*, they are roughly indistinguishable for predictive power — *Markov-Chain* has slightly higher R^2 but also higher error.

Now let us consider the contributions of the *Markov-Chain Sampling* predictors in our models from Table 6. These *variable importances* are calculated as such; for all trees, the sub-sample of observations not used in the building of the tree has prediction accuracy calculated on it. To get the importance of a variable (predictor), a random shuffle is applied to the variable values in the sub-sample. Everything else is kept constant. The prediction accuracy is then tested again on this permuted data. This accuracy decrease is averged over the trees to obtain *variable importance* for a predictor.

Each column is for a random forest model, and the predictors from top to bottom are best to worst. Immediately apparent is the importance of the sampled fitness distribution. In fact, *mean fitness* is the top predictor for all four models. *Funnel-floor* fitness is the second best predictor for both *Markov* models, telling us again the role of sampled fitness levels. *Out-degree* is not an important *Markov-Chain LON* feature for algorithm performance prediction, ranking lowest and second-lowest in the models.

Table 7 reports the rankings for the *Snowball LON Sampling* RF models. Just like in Table 6, *mean fitness* is top-ranked in each. Also important for predicting TS is *fitness correlation*, appearing second. In the middle are the local search metrics collected during the *Snowball LON* process and the *weight disparity*. Ranking lowest is *weight loops*, pertaining to unsuccessful escape attempts from local optima. These findings are unusual: the bulk of LON literature focuses heavily on edge connection patterns and densities in the LON (Herrmann et al., 2016; Thomson et al., 2017; McMenemy et al., 2018) rather than simply the fitness distribution the sample reflects. Our observation corroborates and extends the findings of the preliminary study on LON construction algorithms (Thomson et al., 2019).

Table 8: Model summary statistics when using features from both LON types.

Regression	Response Variable	R^2	MSE
Linear	ILS	0.076	0.003
Linear	TS	0.254	0.026
RandomForest	ILS	**0.972**	**0.000**
RandomForest	TS	**0.991**	**0.000**

5.4 Best of Both Sampling Methods?

Markov-Chain LON Sampling and *Snowball LON Sampling* produce LONs which capture different fitness landscape information. Features to do with edge and node density, as well as those relating to path lengths, are redundant as predictors when calculated on a *Snowball* LON: they are prescribed by the sampling algorithm parameters. A speculation is put forward now that the union of different features from LON types — for the same problem instances — might predict more accurately than using one LON type only. The three best-ranked predictors (according to Section 5.3) for each response variable (*ILS* or *TS*) and each LON method (*Markov* or *Snowball*) are bundled together into single models. The results are in Table 8. Rankings are in Table 9. Table 8 illumi-

Table 9: Predictor rankings for RF models using both LON types.

Predicting ILS	Predicting TS
mean fitness (Snowball LONs)	*mean fitness* (Snowball LONs)
funnel-floor fitness	*funnel-floor fitness*
mean fitness (Markov LONs)	*mean fitness* (Markov LONs)
number optima	maximum HC paths
maximum HC length	*fitness correlation*
mean HC length	edges

nates further that linear regression may not suit LON performance prediction models, with rows 1 and 2 showing weak R^2 values. The lack of specificity suggests they miss non-linear interactions between variables. This is in contrast to the random forest counterparts — rows 3 and 4 — strikingly stronger than the linear models. They account for 97% (ILS) and 99% (TS) of search variance explained. The error rates (see *MSE* column) appear to be low. The R^2 values surpass those seen when using features from one type of LON.

In Table 9, we show the six features ranked by importance in the RF models. As in Section 5.3, the prestige of the fitness features is obvious. Both models share the same three features. The mean sampled *Snowball LON* fitness (*mean fitness* (Snowball LONs)) is top-ranked. In second place is *funnel-floor fitness*, calculated on *Markov LONs* only. In third place is the *Markov LON mean fitness*. As in Section 5.3, the role of the LON connectivity variable (*edges*) is weak, ranking last in its model.

6 Discussion

Markov-Chain LON Sampling has not been validated against 'ground truth' enumerated LONs before. Section 5.1 showed that for small (N=11) QAP instances, both *Markov-Chain LON Sampling* and *Snowball LON Sampling* produced accurate approximation of the number of nodes (local optima) and edges (optima transformations) when comparing to the 'ground-truth' fully enumerated LON. *Markov-Chain LON Sam-*

pling did not find all of them but the quantities correlated with the enumerated LON values. The latter found almost the same number of nodes and edges as the enumerated LON.

Although this is encouraging, there must be caution when tempted by extrapolations to larger problems. Both LON construction algorithms are designed for large problems, and use the according amount of computation. There may be a particular consideration for scaling *Snowball LON Sampling* to larger problems: The process uses short, restricted paths through local optima and is not based on optimisation. Therefore, it is argued *Snowball LON Sampling* may be highly dependent on the random starting solution — the rest of the sample is based around that. In a large fitness landscape, the obtained sample may actually be rather low-quality local optima, far from the promising regions near the global optimum. In future work, the intention is to examine the relationship between LON sampling parameter choices and problem dimension for both algorithms.

Another engaging observation from the sampling validation attempt (Section 5.1) was that generally, sampled LON features explained more search variance than those of the enumerated LONs. This is important because the vast majority of LON research (Ochoa et al., 2008; Daolio et al., 2011; Verel et al., 2011; Herrmann et al., 2016) dissected fully enumerated LONs, even using their features for algorithm performance prediction (Daolio et al., 2012; Herrmann et al., 2018; Thomson et al., 2018). Our results suggest that LON construction algorithms may approximate or infer yet unseen fitness landscapes better than best-improvement exhaustive enumeration of the local optima level. This does, intuitively, conform to logic: LON sampling algorithms use search operators — and sequences of operators — which overlap with search algorithm operators used to optimise QAP instances.

In Section 5.2 we saw that capping the LON construction algorithms to 50,000 fitness function evaluations produces LONs whose features can explain up to around 50% of search variance. The *Snowball LON Sampling* seemed to infer the tabu search response variable better than its competitor; *Markov-Chain Sampling* LONs did slightly better on the ILS response.

The main QAPLIB LONs told us that they have solid predictive power for search algorithms in Section 5.3. Both *Markov-Chain LON Sampling* and *Snowball LON Sampling* seem to produce samples that infer prospective fitness landscapes, with much variance in search being explained with the features.

Sections 5.3 and 5.4 showed us that random forest trees yield more promising regression models than linear regression. This suggests the use of RF in future algorithm performance prediction models using fitness landscape features, in pursuit of capturing complex variable interactions.

Another aspect of the results was the apparent importance of the fitness levels sampled by the LON construction algorithms, seen in Sections 5.3 and 5.4. Metrics about the distribution were repeatedly present in the top two predictors for random forest models. This particular phenomenon brings more weight to the same observation made by Thomson et al. (2018) — the present study uses three times more instances, and more than double the included problem dimensions.

The strongest models we saw were when the best features from *Markov-Chain* LONs were combined with the best features from the *Snowball* LONs in Section 5.4, accounting for 97% (ILS) and 99% (TS) of search variances. The suggestion is therefore combining features from both LON construction algorithms as a 'super-sample' in predictive models. In the future, the intention is to implement a combined LON algorithm

which draws on both the exploitation of the *Snowball* algorithm and the exploration of the *Markov* algorithm.

There are certainly limitations to the approach and indeed the results presented. The results are valid but currently hold only for our particular search operator choices, configurations of LON construction algorithms, choices of QAP heuristic search algorithms, choice for quantifying search success, chosen neighbourhood function, and the features taken from the LONs. There is also the issue of the non-determinism in the sampling algorithms.

7 Conclusions

Local Optima Network (LON) construction algorithms have been scrutinised to gain information about their ability to infer future fitness landscapes. The two most recent LON sampling algorithms were studied, and they were named *Markov-Chain LON Sampling* and *Snowball LON Sampling*. The aim was to compare them and assess the quality of the resultant LON samples for predicting performance on the problems. We also used an algorithm for exhaustively enumerating LONs. All were applied to the Quadratic Assignment Problem, but the frameworks could easily be modified to a different problem in the manner of metaheuristic framework. The QAP instance set included both benchmark and generated problems.

The study began with validation of LON samples by comparison with 'ground truth' (the enumerated LON for the same instance). Then *Markov-Chain LON Sampling* and *Snowball LON Sampling* were given 50,000 fitness evaluations to produce LONs of QAPLIB instances. After that *Markov-Chain LON Sampling* and *Snowball LON Sampling* were given default computation (according to the algorithms specification by the LON algorithm authors). A large set of QAPLIB instances up to N=128 had fourteen LONs mapped in this way. Finally, the three most important features for each type of LON sample were taken and joined into one model.

The results suggested that for optimisation predictions made by LON features, random forest trees better capture the non-linear effects between variables in the fitness landscape. Repeatedly, the apparent key role of sampled fitness levels in explaining search algorithm variance was seen. A lot of LON literature focuses on edge connectivity patterns and not the fitness distribution amongst local optima — is this misguided? A surprising result was that the enumerated LON, when compared to sampled LON, had less predictive power. An intuitive thought is that a better prediction comes from more extensive fitness landscape information. This is not the case here. An argument is made that this stems from LON sampling algorithms sharing similar operators — or indeed sequences of operators — with heuristics used in QAP research.

This study ends with a note that there is an abundant field of work waiting to be done in this area, and we are excited to explore it. In particular, we acknowledge that our results are of course dependent on our algorithm configurations, neighbourhood function choice, choices for search algorithms, choice of problem domain, and so on. The pursuit of different choices is the next endeavour.

References

Applegate, D., Cook, W., and Rohe, A. (2003). Chained lin-kernighan for large traveling salesman problems. *INFORMS Journal on Computing*, 15(1):82–92.

Bożejko, W., Gnatowski, A., Niżyński, T., Affenzeller, M., and Beham, A. (2018). Lo-

cal optima networks in solving algorithm selection problem for tsp. In *International Conference on Dependability and Complex Systems*, pages 83–93. Springer.

Burkard, R. E., Karisch, S. E., and Rendl, F. (1997). Qaplib — a quadratic assignment problem library. *Journal of Global optimization*, 10(4):391–403.

Chicano, F., Ochoa, G., Whitley, D., and Tinós, R. (2018). Enhancing partition crossover with articulation points analysis. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 269–276. ACM.

Daolio, F., Tomassini, M., Vérel, S., and Ochoa, G. (2011). Communities of minima in local optima networks of combinatorial spaces. *Physica A: Statistical Mechanics and its Applications*, 390(9):1684–1694.

Daolio, F., Verel, S., Ochoa, G., and Tomassini, M. (2012). Local optima networks and the performance of iterated local search. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, pages 369–376. ACM.

Doye, J. P., Miller, M. A., and Wales, D. J. (1999). The double-funnel energy landscape of the 38-atom lennard-jones cluster. *The Journal of Chemical Physics*, 110(14):6896–6906.

Fieldsend, J. E. (2018). Computationally efficient local optima network construction. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1481–1488. ACM.

Goodman, L. A. (1961). Snowball sampling. *The Annals of Mathematical Statistics*, pages 148–170.

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

Hernando, L., Daolio, F., Veerapen, N., and Ochoa, G. (2017). Local optima networks of the permutation flowshop scheduling problem: Makespan vs. total flow time. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 1964–1971. IEEE.

Herrmann, S. (2016). Determining the difficulty of landscapes by pagerank centrality in local optima networks. In *Evolutionary Computation in Combinatorial Optimization*, pages 74–87. Springer.

Herrmann, S., Ochoa, G., and Rothlauf, F. (2016). Communities of local optima as funnels in fitness landscapes. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference*, pages 325–331. ACM.

Herrmann, S., Ochoa, G., and Rothlauf, F. (2018). Pagerank centrality for performance prediction: the impact of the local optima network model. *Journal of Heuristics*, 24(3):243–264.

Iclanzan, D., Daolio, F., and Tomassini, M. (2014). Data-driven local optima network characterization of qaplib instances. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pages 453–460. ACM.

Lawler, E. L. (1963). The quadratic assignment problem. *Management science*, 9(4):586–599.

Liu, J., Abbass, H. A., and Tan, K. C. (2019). Problem difficulty analysis based on complex networks. In *Evolutionary Computation and Complex Networks*, pages 39–52. Springer.

McMenemy, P., Veerapen, N., and Ochoa, G. (2018). How perturbation strength shapes the global structure of tsp fitness landscapes. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 34–49. Springer.

Nakagawa, S. and Schielzeth, H. (2013). A general and simple method for obtaining r2 from generalized linear mixed-effects models. *Methods in Ecology and Evolution*, 4(2):133–142.

Ochoa, G. and Herrmann, S. (2018). Perturbation strength and the global structure of qap fitness landscapes. In *15th International Conference, Coimbra, Portugal, September 8–12, 2018, Proceedings, Part II*, pages 245–256.

Ochoa, G., Tomassini, M., Vérel, S., and Darabos, C. (2008). A study of nk landscapes' basins and local optima networks. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 555–562. ACM.

Ochoa, G. and Veerapen, N. (2016). Deconstructing the big valley search space hypothesis. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 58–73. Springer.

Ochoa, G. and Veerapen, N. (2018). Mapping the global structure of tsp fitness landscapes. *Journal of Heuristics*, 24(3):265–294.

Ochoa, G., Veerapen, N., Daolio, F., and Tomassini, M. (2017). Understanding phase transitions with local optima networks: Number partitioning as a case study. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 233–248. Springer.

Simoncini, D., Barbe, S., Schiex, T., and Verel, S. (2018). Fitness landscape analysis around the optimum in computational protein design. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 355–362. ACM.

Stadler, P. F. (2002). Fitness landscapes. In *Biological evolution and statistical physics*, pages 183–204. Springer.

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

Taillard, É. (1991). Robust taboo search for the quadratic assignment problem. *Parallel computing*, 17(4-5):443–455.

Tayarani-N, M.-H. and Prügel-Bennett, A. (2016). An analysis of the fitness landscape of travelling salesman problem. *Evolutionary computation*, 24(2):347–384.

Thomson, S. L., Daolio, F., and Ochoa, G. (2017). Comparing communities of optima with funnels in combinatorial fitness landscapes. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '17, pages 377–384, New York, NY, USA. ACM.

Thomson, S. L., Ochoa, G., and Verel, S. (2019). Clarifying the difference in local optima network sampling algorithms. In *European Conference on Evolutionary Computation in Combinatorial Optimization (Part of EvoStar)*, pages 163–178. Springer.

Thomson, S. L., Verel, S., Ochoa, G., Veerapen, N., and McMenemy, P. (2018). On the fractal nature of local optima networks. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 18–33. Springer.

Tomassini, M., Verel, S., and Ochoa, G. (2008). Complex-network analysis of combinatorial spaces: The n k landscape case. *Physical Review E*, 78(6):066114.

Veerapen, N., Ochoa, G., Tinós, R., and Whitley, D. (2016). Tunnelling crossover networks for the asymmetric tsp. In *International Conference on Parallel Problem Solving from Nature*, pages 994–1003. Springer.

Verel, S., Daolio, F., Ochoa, G., and Tomassini, M. (2011). Local optima networks with escape edges. In *International Conference on Artificial Evolution (Evolution Artificielle)*, pages 49–60. Springer.

Verel, S., Daolio, F., Ochoa, G., and Tomassini, M. (2018). Sampling local optima networks of large combinatorial search spaces: The qap case. In Auger, A., Fonseca, C. M., Lourenço, N., Machado, P., Paquete, L., and Whitley, D., editors, *Parallel Problem Solving from Nature – PPSN XV*, pages 257–268. Springer International Publishing.