The Local Optima Level in Chemotherapy Schedule Optimisation

Sarah L. Thomson 0000-0001-6971-7817 and Gabriela Ochoa 0000-0001-7649-5669

Computing Science and Mathematics, University of Stirling, Stirling, UK
{s.l.thomson,gabriela.ochoa}@stir.ac.uk

Abstract. In this paper a multi-drug Chemotherapy Schedule Optimisation Problem (CSOP) is subject to Local Optima Network (LON) analysis. LONs capture global patterns in fitness landscapes. CSOPs have not previously been subject to fitness landscape analysis. We fill this gap: LONs are constructed and studied for meaningful structure. The CSOP formulation presents novel challenges and questions for the LON model because there are infeasible regions in the fitness landscape and an unknown global optimum; it also brings a topic from healthcare to LON analysis. Two LON Construction algorithms are proposed for sampling CSOP fitness landscapes: a Markov-Chain Construction Algorithm and a Hybrid Construction Algorithm. The results provide new insight into LONs of highly-constrained spaces, and into the proficiency of search operators on the CSOP. Iterated Local Search and Memetic Search, which are the foundations for the LON algorithms, are found to markedly out-perform a Genetic Algorithm from the literature.

Keywords: Combinatorial Fitness Landscapes, Local Optima Networks, Search Space Analysis

1 Introduction

Local Optima Networks (LONs) [1] are used to study fitness landscapes. Analysis of them provides insight into how optimisation problems and search algorithms interact together. LONs capture *global* patterns at the Local Optima Level (LOL) in landscapes and have mostly been extracted for benchmark combinatorial optimisation problems such as NK Landscapes [1–3], the Quadratic Assignment Problem (QAP) [4–6], and the Travelling Salesman Problem (TSP) [7–9].

Studies in non-benchmark problem domains have been sparse and have consisted of computational protein modelling [10] and feature selection [11]. These were steps towards bringing LON analysis to unmapped real-world problems. This type of case study, demonstrating LON efficacy, is needed for convincing possible industry collaborators. Large and highly-constrained problems should ideally be used in case studies (by 'large' we mean hundreds of dimensions), in pursuit of simulating environments typical of real-world optimisation problems.

Chemotherapy Schedule Optimisation Problems (CSOPs) [12] have been the subject of several research papers in evolutionary computation [13–17]. One instance was formulated to reflect real-life chemotherapy drug response closely [13] and the tumour shrinkage model used in the fitness function has been subject to extensive clinical testing [18]. The instance, alongside other CSOP formulations, has not been subject to fitness landscape analysis (although some authors have made passing remarks about CSOP landscapes [19, 20]).

We conduct a first fitness landscape analysis on CSOP, focussing on the LOL with the use of LONs. Two LON Construction algorithms are proposed — the first has Iterated Local Search (ILS) as its foundation; the second has Memetic Search (MS) as the foundation. LONs are then produced and their attributes and fitness distributions are compared. A study of the feasibility trajectories in the LONs is also presented. Later on, algorithm performance results suggest our search algorithms (ILS and MS) outperform a GA from the literature for the CSOP. In summary, the present work contributes in the following ways:

1. First fitness landscape analysis of CSOP, lending to new insights of the problem interacting with search operators;
2. The presence of infeasible solutions in the landscapes is new to LON research;
3. Two LON Construction algorithms for the CSOP are proposed (which can also be easily applied to an arbitrary binary-encoded problem).
4. Two search algorithms are offered which outperform a GA from the literature (ILS and MS; a separate MS has been used on a CSOP formulation before but with different fitness function, constraints, and solution encoding [19]).

1.1 Background

We use a multi-drug CSOP which was initially formulated and described in 1998 [13] and then further studied in later research [14–17, 21]. As asserted in the original paper, a multi-drug CSOP can have a binary representation where each gene, i, is set iff a particular concentration of a particular drug (of number n) is administered at a particular time interval (t, from within defined time intervals). As suggested in the literature [21], we set the number of drugs $n = 10$ and the number of time intervals for doses, t, also at 10. There are four allowed concentrations for each drug, $p = 4$, giving each binary solution a length of 400, i.e. $n \times t \times p$. The number of possible solutions, and the size of the configuration space, is extremely large at 2^{ntp} i.e. 2^{400}.

1.2 Fitness Function

We consider *curative* chemotherapy treatment here, meaning tumour eradication is the aim. This is the primary (and only) objective. For this single-objective case, fitness is calculated with respect to the chemotherapy schedule *minimising* tumour size (in number of cells). This is done through *maximising* the combined effect of drugs in the schedule against the tumour. In considering the tumour's shrinkage response, a mathematical function is needed. The most popular model in the literature is called the *Gompertz Growth Model* [18], which has a linear cell-loss effect and has been validated by significant clinical experiments. The formula is given in Equation 1:

$$\frac{dN}{dt} = N(t)\left(\lambda \ln\left(\frac{\theta}{N(t)}\right) - \sum_{j=1}^{d} k_j \sum_{n}^{i=1} C_{ij}\left\{H(t-t_i) - H(t-t_{i+1})\right\}\right) \qquad (1)$$

with the components as follows: $N(t)$ is the cancerous cell count at time interval t; λ and θ are parameters pertaining to tumour growth; $H(t)$ is the Heaviside step function; k_j denotes the efficacy of chemotherapy drugs; and C_{ij} is the concentration levels of the drugs administered.

The actual fitness function is quite complex, including penalties based on feasibility distances, and in the interest of space we refer the interested reader to a comprehensive description [13] (pp. 106-107). In essence, initial fitness is calculated with respect to the total impact on the tumour for the treatment schedule. Individual impacts for each drug are known. The objective is to maximise the combined impact of all the drugs in the schedule (at the specified concentrations, and at the specified time-slots). The *maximisation* of this will *minimise* the tumour.

Following the drug impact fitness calculation, the solution is checked for constraint violations and the fitness is penalised accordingly (see [13] for details). Any violation will result in a fitness below zero. A feasible solution has fitness zero or above.

The constraints are as follows: the tumour is not allowed above a particular size; the maximum cumulative dose of drugs cannot exceeded the specified limits for each individual drug; and the limit on toxic chemotherapy side-effects cannot be exceeded (for each time interval). In all cases the magnitude of the violation is captured through proportional subtraction from the fitness sum.

Mathematically the fitness function is subject to these constraints:

1. Maximum allowable *cumulative* C_{cum} dosage for each drug:

$$g_1(c) = \left\{ C_{cum\,j} - \sum_{i=1}^{n} C_{ij} \;\geq\; 0 \vdots \forall j \in \overline{1,d} \right\} \qquad (2)$$

2. Maximum allowable size of the tumour, i.e. number of cancerous cells, N:

$$g_2(c) = \left\{ N_{max} - N(t_i) \;\geq 0 \vdots \forall i \in \overline{1,n} \right\} \qquad (3)$$

3. A threshold for the known toxic side-effects of using multiple drugs in chemotherapy treatment:

$$g_3(c) = \left\{ C_{s-e_k} - \sum_{j=1}^{d} \eta_{kj} C_{ij} \;\geq\; 0 \vdots \forall i \in \overline{1,n}, \forall k \in \overline{1,m} \right\} \qquad (4)$$

In the constraint seen in Equation 4, the variables η_{kj} are the known possibility of harming the k^{th} organ (for example, the heart) through administering the j^{th} drug.

1.3 Evolutionary Search Algorithms

Evolutionary algorithms have been used with success for CSOPs; in particular, Genetic Algorithms (GAs) have dominated [13, 21–23], although other approaches have been utilised, such as Estimation of Distribution Algorithms [15, 17]; Simulated Annealing variants [23, 24]; Memetic Algorithm (MA) [19]; and Evolutionary Strategies [23, 25]. A GA from the literature [17] is used as the foundation for the *Hybrid* LON Construction algorithm proposed here (detailed later in Section 2.2) and is also used later on in conducting optimisation on the problem to collect search difficulty information.

2 Methodology

This section describes the LON Construction algorithms proposed for studying CSOP fitness landscapes. Our aim is examining the topological features forming when optimisation search operators are moving on the CSOP configuration space. The particular focus is on *global-scale* local optima connectivity patterns.

2.1 Markov-Chain LON Construction Algorithm

To align with existing LON Construction algorithms for benchmark domains such as TSP and QAP [6, 26, 27] we instrument an algorithm using Iterated Local Search (ILS) as the vehicle. ILS is naturally suited to constructing LONs: each iteration identifies a transformation between local optima and this can straightforwardly be added as an edge to a LON. We refer to the ILS-driven LON Construction algorithm as *Markov-Chain* LON Construction — to avoid confusion, because ILS is also used later on to collect difficulty information about the CSOP.

Markov-Chain LON Construction tracks thirty independent ILS runs, which begin from random solutions. The local search is best-improvement and uses one-flip neighbourhood. Perturbation flips thirty bits. Improving local optima are always accepted; 10% of the time, worsening local optima are accepted too. All accepted local optima are added as LON nodes and the transformation is logged as a LON edge (if the edge exists already, the weight is incremented). Runs terminate after 1000 iterations. Parameters were chosen in response to observations about preliminary runs.

Nodes and edges from the thirty runs are joined together to form a single LON for the problem. Our initial intention was to mirror parameter choices in previous LON Construction works [27] but those choices were for much smaller search space sizes and the computation was therefore more feasible for their circumstance. The complete process for *Markov-Chain* LON Construction is provided in Algorithm 1.

Algorithm 1 Markov-Chain LON Construction

```
Search space $S$, Fitness function $f$,
Perturbation strength $k$, Stopping threshold $t$, Number of runs $r$
$runs \leftarrow 0$
repeat
    Choose initial random solution $s_0 \in S$
    $l \leftarrow$ LocalSearch($s_0$)
    $i \leftarrow 0$
    repeat
        $s' \leftarrow$ Perturbation($l_1$, $k$)
        $l_2 \leftarrow$ LocalSearch($s'$)
        if $f(l_2) \leq f(l_1)$ then
            $l_1 \leftarrow l_2$
        end if
        $LON = LON +$ nodes($l_1, l_2$)
        $LON = LON +$ edge($l_1 \longrightarrow l_2$)
        $i \leftarrow i + 1$
    until $i \geq t$ return $l$
    $runs \leftarrow r + 1$
until $runs \geq r$
```

2.2 Hybrid LON Construction Algorithm

Genetic Algorithms (GAs) have been successful in finding good approximate solutions in CSOP [16]; it follows that a GA is a reasonable foundation for CSOP LON Construction. When LON Construction algorithms share operators with successful heuristics, the constructed LONs should *infer* future landscapes that might be induced during genuine optimisation.

Our LON Construction algorithm originates from a generational GA from the literature [14]. By definition LONs contain only local optima. GAs, of course, do not guarantee local optima in the population, which necessitates the addition of local search to the algorithm, resulting in a Memetic Search (MS). The Memetic Search-driven LON tracking process is hereafter referred to by *Hybrid* LON Construction, to differentiate from a MS used later for collecting problem difficulty information.

The algorithmic process for creating the LON is as follows. The algorithm runs for 100 generations; at each generation the fittest 10% of offspring are subject to local search to produce local optima. To deem a node a local optimum, one-flip best-improvement hill-climbing is applied for 100 iterations. The nodes are added to the LON and are put into the next generation. The set of local optima are then deterministically recombined with one another. The offspring are possibly mutated according to the mutation rate, before being subject to local search. All four local optima (parent one and two, child one and two) are then added as nodes to the LON. Similarly, four edges are added to the network: from parent one to child one; parent one to child two; parent two to child one; and parent two to child two.

Algorithm 2 Hybrid LON Construction: Part 1

```
procedure Hybrid LON Construction(population size $ps$, generations $g$, percent fittest individuals $pf$, mutation probability $mp$, crossover probability $cp$, length evolution path $lp$, search space $S$, fitness function $f$)
    $LON = \emptyset$                                  ▷ local optima network
    $P =$ randomPopulation($S$, $ps$)                  ▷ Initial random population
    $fit =$ SelectFittest($P$, $pf$, $f$)              ▷ fittest individuals
    $iterations = 0$                                   ▷ counter for generations completed
    repeat
        $P =$ GeneticProcess($P$, $ps$)
        $fit =$ SelectFittest($P$, $pf$)
        for $sol \in fit$ do
            $sol =$ HillClimb($sol$)
        end for
        for $mom \in fit$ do
            for $dad \in fit$ do
                $child_1, child_2 =$ MemeticEvolution($mom, dad$)
                $iterations = iterations + 1$
            end for
        end for
    until $iterations \geq g$
    for $mom \in LON$ do
        for $dad \in LON$ do
            $steps = 0$
            repeat
                $child_1, child_2 =$ MemeticEvolution($mom, dad$)
                $mom, dad = child_1, child_2$
                $steps = steps + 1$
            until $steps \geq lp$
        end for
    end for
end procedure
```

After all generations are complete, LON nodes undergo another evolutionary process. For each pairwise combination of nodes, the following is repeated ten times: the solutions are deterministically recombined with one another; the offspring are probabilistically subject to mutation; the offspring are subject to local search. They are added as LON nodes, and transformations from parent to child are added as edges. After this, the locally-optimised offspring become the parents for the next iteration of the same process. The steps repeat ten times. In this way, each pair of original LON nodes (from the 100-generation MS) are the ancestors in a ten-generation evolutionary trajectory. This was a deliberate design choice to facilitate LONs containing sequences of evolution for local optima. Without this, the LON would consist of many isolated pairs of nodes and would be difficult to study for meaningful structure. The complete process for the *Hybrid* LON Construction algorithm is shown in Algorithms 2 and 3.

Algorithm 3 Hybrid LON Construction: Part 2

```
procedure GeneticProcess($P$, $ps$)
    repeat
        $mom$, $dad$ = Selection($P$)
        if $cp$ then
            $child_1, child_2$ = Crossover($mom$, $dad$)
        end if
        if $mp$ then
            $child_1, child_2$ = Mutation($child_1, child_2$)
        end if
        $P[mom] = child_1$
        $P[dad] = child_2$
    until iterations $\geq ps/2$
end procedure

procedure MemeticEvolution($mom, dad$)
    $child_1, child_2$ = Crossover($mom, dad$)
    if $mp$ then
        $child_1, child_2$ = Mutation($child_1, child_2$)
    end if
    $child_1, child_2$ = HillClimb($child_1, child_2$)
    $LON = LON$ + nodes($child_1, child_2$)
    $LON = LON$ + edges($\{mom \longrightarrow child_1\}$, $\{mom \longrightarrow child_2\}$, $\{dad \longrightarrow child_1\}$, $\{dad \longrightarrow child_2\}$)
    return $child_1, child_2$
end procedure
```

3 Visualisations

Visual analysis of LONs can provide an abstracted view of the Local Optima Level, which is a multi-dimensional complex system. Sometimes, patterns observable in visual analysis help to explain search algorithm performance on the associated combinatorial problem.

Markov-Chain LON Construction and *Hybrid* LON Construction algorithms produce networks with thousands of nodes. For meaningful visualisation, pruned sub-networks are constructed. The 'elite' nodes of the LONs are chosen for this. For the *Markov-Chain* LON, these are nodes in the top 2% of the fitness distribution. The *Hybrid* LON has more nodes, so only the top 0.05% are visualised. It follows that this lifts the veil on the most promising regions reached by the algorithms.

Figure 1 shows plots for two LONs of the same CSOP. The top Figure is the LON constructed by the *Markov-Chain* method; on the bottom was constructed with the *Hybrid* method. Edges encode sequences of search operations. On the

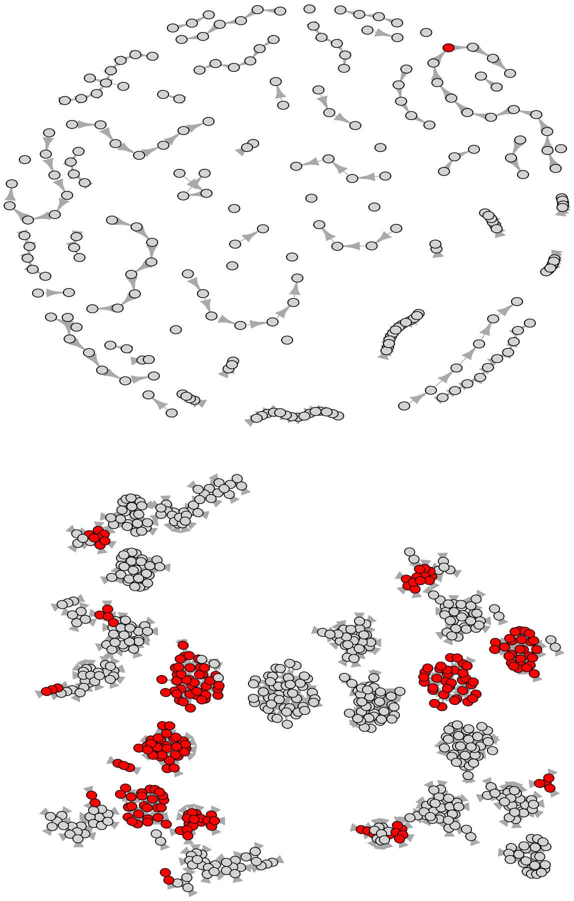

Fig. 1. Top 2% of local optima which were sampled using *Markov-Chain* LON Construction (top) and top 0.05% sampled during *Hybrid* LON Construction (bottom). Pseudo-global optima (i.e. the best in that particular sample for the purposes of this visualisation) are shown in red; all other local optima are grey. For the *Markov-Chain* LON, the highest fitness is 1.707677. For the *Hybrid* LON, it is 1.707826.

higher LON, the sequence is perturbation $\longrightarrow$ local search; on the bottom, recombination $\longrightarrow$ probabilistic random mutation $\longrightarrow$ local search. Nodes with the highest fitness in that sample are red; all other local optima are grey.

Examining the Figure, *Markov-Chain* is the sparser network of the two. There are neat sequences of local optima and nodes typically have one incoming edge and one outgoing edge. The sequences are separate, in that they do not have bridges connecting them. The highest-fitness node is located within a sequence. The visual analysis of this implies that this fitness would only be reached if the search arrived by happenstance on that particular sequence of local optima.

The *Hybrid* LON is denser and instead of linear sequences, clusters of nodes are seen. Some clusters are connected to other clusters and some are isolated. Many distinct solutions with the pseudo-optimal fitness are found by the *Hybrid* (look at the red nodes) and these are found in different clusters. The presence of clusters instead of linear sequences hints that at lower fitness levels (not shown) the clusters would be larger and more opportunity to connect to a pseudo-optimum would be found when comparing to the *Markov-Chain* LON.

4 Experimental Setup

4.1 *Markov-Chain* LON Construction: Details

As stipulated in Section 2.1, *Markov-Chain* LON Construction algorithm is an Iterated Local Search (ILS) framework. As such, local search handles intensification and perturbation mechanism contributes diversification. The local search uses a bit-flip operator and best-improvement as a pivot rule. A solution is deemed a local optimum at the end of 100 iterations. Perturbation is 30 bit-flips and improving local optima are always accepted. Deteriorating local optima are accepted 10% of the time. Runs terminate after 1000 iterations. Thirty independent runs are conducted, with each accepted local optimum added as a LON node and each transformation between two local optima added as a LON edge. The parameters are shown in Table 1.

Table 1. *Markov-Chain* LON Construction parameter settings

Parameter	value
Local iterations	100
Global iterations	1000
Pivot rule	Best
Local search	1 bit-flips
Perturbation	30 bit-flips
Number of runs	30

Iterated Local Search. The ILS process from *Markov-Chain* LON Construction is modified (without any LON logging) and is proposed for optimisation of CSOP. We use it in collecting algorithm performance information. The algorithmic setting remains the same except the best-improvement rule changes to the best of 100.

4.2 Hybrid LON Construction: Details

Hybrid LON Construction is instrumented on top of a competitive GA for the domain [13]. A previous study using statistical inference found that only two GA parameters were significant on this CSOP when solutions are binary-encoded: crossover probability $\phi\prime$ and mutation probability $\phi\prime\prime$ [21]. Our values for those parameters are the ones they recommended ($\phi\prime$=0.614, $\phi\prime\prime$=0.198); the others are from a related study [14] (which used integer encoding for the problem), in the absence of reported values in the binary-encoded study. A random starting population of 76 individuals, all binary strings with $n = 400$, is created. Elitism is implemented for the fittest two individuals; the selection method is linear roulette-wheel (parents are selected with probability proportionate to their fitness ranking); selection pressure is seven; and there are six points of crossover, with the crossover type being uniform. We added local search, making the algorithm memetic. The local search was best-improvement, using single bit-flips, and for 100 iterations. This is applied to the best 10% of individuals at each generation. Those individuals are added as LON nodes, recombined, and the trajectories from parent to child are taken as LON edges. After 100 generations, pairwise combinations of LON nodes are recursively recombined 10 times: offspring from the first recombination are subject to local search and then become the parents for the next. Nodes and edges are added to the LON during this process.

Memetic Search. A variant of the MS framework described in the previous Section (without any LON logging) is also proposed here for optimisation on the CSOP. In our study we use it to collect algorithm performance information. The percentage of individuals locally optimised at each generation becomes 5%; the best-improvement local search becomes first-improvement; and the local search operator becomes ten bit-flips instead of one. All parameter settings for the GA component remain the same. 100 generations are allowed.

4.3 Unknown Global Optimum

For this problem the global optimum is not known. In previous LON research, there has always been a known optimum fitness. To simulate this for our problem, we conduct several runs of the MS and ILS and take the highest obtained fitness across all runs to be the pseudo-optimal fitness. This value is 1.71.

5 Results

5.1 The Hybrid LON

The *Hybrid* LON Construction network has 124,497 nodes and 1,264,500 edges, giving an edge-to-node ratio of 10:1. The average fitness is 0.909698, which at above zero is a feasible solution and is around 53% of the pseudo-optimal fitness stipulated in Section 4.3. The maximum fitness is 1.707826, which is within 0.001% of the pseudo-optimal fitness. The minimum fitness (of a local optimum in the sample) is -106.717. There are 217 different solutions with the pseudo-optimal fitness. The vast majority — 93.6% — of the local optima are feasible solutions.

Around 14.5% of edges in the LON represent no fitness change; 43.6% are improving fitness; and 41.8% have deteriorating fitness.

The *assortativity* coefficient of a network is the Pearson's correlation for the degrees of connected nodes. In the *Hybrid*-constructed LON, it stands at 0.794687. This implies that it is likely for a node to connect to nodes which have similar degree.

The median degree for a node in the LON is 10; the mean is 20.31; the 75% quantile is 16; and the maximum is very large at 179,154. Most nodes have relatively low degree ($\leq$ 16) and only 0.1% of nodes have degree $\geq$ 241. The presence of a single node with excessively high degree (179154) hints at a hub-and-spoke system being present in a section of the LON.

5.2 The Markov-Chain LON

The *Markov-Chain* LON Construction models an adaptive walk through the LOL. There are 11,393 nodes and 209,489 edges in the sample, for an edge-to-node ratio of approximately 20:1. The average sampled fitness is 0.638913 — around 37% of the pseudo-optimal fitness. This is noticeably lower than the average fitness in the *Hybrid* LON. The maximum fitness is 1.707677, which is lower than both the pseudo-optimal fitness and the maximum fitness in the *Hybrid* LON but is still within 0.002% of the pseudo-optimal value. The minimum is -61.6745, which is approximately twice as fit as the lowest in the *Hybrid* LON. This makes sense given the unguided nature of selection in the *Hybrid* algorithm compared to the guided walk of the *Markov-Chain* process.

In the LON, around 64% of edges are deteriorating (that is, they orient towards a worse fitness); 26% are improving; and around 9% direct towards equal fitness. The majority are deteriorating even though deteriorating moves are only accepted 10% of the time. This fact hints at the scarcity of improving moves on the local optima level manifesting under these operators. Let us compare the percentages with those present in the *Hybrid* LON (43.6% improving and 41.8% deteriorating, as we recall); a judicious conclusion is that the recombination $\longrightarrow$ local search sequence of the *Hybrid* algorithm has more *evolvability* potential on the LOL than the perturbation $\longrightarrow$ local search sequence of the *Markov-Chain* algorithm.

The assortativity coefficient is 0.996704, which stipulates that nodes are highly likely to be connected to nodes which have the same degree as them. This is evidence against the presence of a 'hub-and-spoke' network structure in this LON because that phenomenon is defined by heterogeneous degree distribution. The median degree in the LON is 34; the mean is close by at around 37; the 75% quantile is 52; and the maximum degree is 526. Only 0.01% of nodes have degree $\geq$ 128. The range of values in the degree distribution is much less extreme than was present in the *Hybrid* LON.

5.3 A Study of Feasibility in LONs

The existence of infeasible solutions in CSOP fitness landscapes brings new possibilities for the features calculated from the LONs. One consideration is the proportion of LON nodes which are infeasible. In the *Hybrid* LON Construction object, 93.6% of nodes are feasible (meaning they have fitness above 0.0000); the *Markov-Chain* LON Construction object has 86.2% feasible nodes. This implies that the former more heavily exploits feasible regions at the level of local optima. Another detail that can be studied for the LONs is the notion of *feasibility gradient*. This is the change in fitness feasibility that a LON edge encodes. An edge could be oriented from an infeasible local optimum towards a feasible local optimum, which is a desirable situation. The distribution of feasibility gradients in the LON therefore captures the ability of the LON algorithm to escape infeasible regions.

Table 2. Percentage of *Markov-Chain* LON edges in terms of feasibility gradient

orientation	percentage
infeasible ⟶ feasible	≈77%
infeasible ⟶ infeasible	≈14%
feasible ⟶ feasible	≈5%
feasible ⟶ infeasible	≈4%

Table 3. Percentage of *Hybrid* LON edges in terms of feasibility gradient

parameter	value
feasible ⟶ feasible	≈84.8%
infeasible ⟶ feasible	≈7.7%
infeasible ⟶ infeasible	≈4.1%
feasible ⟶ infeasible	≈3.4%

In Table 2 are indications of the feasibility gradients (in percentage terms) of the edges of the sampled *Markov-Chain* LON. These must be viewed with the consideration that the algorithm used to construct the edges always accepts improving local optima, but also accepts deteriorating local optima 10% of the time. Encouragingly, the large majority (77%) of edges orient from infeasible to feasible local optima. That implies the operator sequence often succeeds in traversing portals out of infeasible regions. Transformations from feasible ⟶ feasible are much fewer at approximately 5% of total edges. This perhaps implies that the operator sequence is not great at exploiting within the feasible regions in the search space.

Table 3 shows the feasibility gradient percentages seen in the *Hybrid* LON. Here a vast majority (84.8%) of the orientations are from feasible ⟶ feasible. This hints the operator sequence is proficient at intensification within promising areas in the search space. The percentage of directions from infeasible ⟶ feasible is small, which could also be important — maybe the algorithm struggles to escape infeasible areas. It could be, however, that this small percentage is born from the fact that the number of infeasible nodes in the network is low. A surprisingly low percentage (3.4%) lead from feasible ⟶ infeasible solutions. This is interesting, because there is no acceptance condition for nodes during the construction. It seems recombining already-fit solutions before refining the offspring with local search results in fit solutions.

5.4 Newly Proposed Optimisation Algorithms

As described in Sections 4.1 and 4.2 we propose and use modified versions of our ILS and MS frameworks to conduct optimisation on the CSOP and compare algorithm performances, alongside the GA from the literature [21]. Tables 4 and 5 summarise distributions for the obtained fitness (averaged over 100 runs) for each algorithm variant. Table 4 displays algorithm results when not enforcing a fitness evaluation budget; Table 5 shows results from the versions which were budgeted 50 000 fitness evaluations. Each row is an algorithm variant. Indications of the variant are found in the *algorithm* and *seeded* columns. By *seeded*, we mean that for these runs a specific solution was seeded into the algorithms. The solution was not chosen due to good fitness (in fact, the fitness is infeasible and heavily-penalised) but rather to provide the same solution across algorithms for 100 runs. In the case of GA and MA, this solution was one individual in the starting populations; for the ILS, it served as the starting solution. As asserted in Section 4.3, we assume the pseudo-optimal fitness value of 1.71 for the purposes of this study.

Table 4. Averaged obtained fitness over 100 runs of the algorithms. In the case of the EAs, this is the best fitness in the population. No computational budget is specified.

algorithm	seeded	minimum	1st quantile	median	mean	3rd quantile	maximum
ILS	no	-25.43	-7.39	-2.09	-4.49	-0.48	**1.66**
GA	no	-71 556.39	-97.64	-32.15	-3738.60	-11.67	**0.40**
MS	no	**1.46**	**1.70**	**1.71**	**1.70**	**1.71**	**1.71**
ILS	yes	-62.54	-6.61	-1.87	-4.86	-0.31	**1.67**
GA	yes	-137 600.69	-212.50	-32.79	-4084.30	-7.55	-0.19
MS	yes	**1.45**	**1.70**	**1.71**	**1.68**	**1.71**	**1.71**

Across budgeted and non-budgeted runs, seeded and unseeded runs, the Genetic Algorithm (GA) is definitely the least competent at obtaining fit (or even feasible) solutions. This can be seen by, for example, comparing the median of

Table 5. Averaged obtained fitness over 100 runs of the algorithms. In the case of the EAs, this is the best fitness in the population. The computational budget for each run is 50 000 function evaluations.

algorithm	seeded	minimum	1st quantile	median	mean	3rd quantile	maximum
ILS	no	-55.89	-10.22	-4.26	-7.38	-0.64	**1.66**
GA	no	-235 584.20	-2538.16	-75.86	-12 476.17	-32.63	-4.07
MS	no	**1.15**	**1.46**	**1.66**	**1.59**	**1.68**	**1.70**
ILS	yes	-46.84	-6.25	-1.96	-4.18	**0.15**	**1.44**
GA	yes	-271 039.28	-4572.31	-63.20	-10 765.08	-27.74	**1.19**
MS	yes	**1.31**	**1.45**	**1.67**	**1.60**	**1.68**	**1.70**

the GA rows with the median of the MS or ILS rows in either of Table 4 or Table 5. Sometimes the difference in fitness is several orders of magnitude — see for example the algorithm comparison available *minimum* columns.

The MS performs by far the best of the three algorithms. In all cases (budgeted and unbudgeted, seeded and unseeded), 100% of the runs ended with a feasible fitness found in the population. That is shown in the third and sixth rows of both tables. The best ILS runs (the *maximum* rows) ended with a desirable fitness although the majority ended with infeasible fitness. It is of note, however, that the distributions comprise fitness values which are ‘almost’ feasible in many cases. The deduction seems to be that the success (or lack thereof) of ILS on this CSOP depends on the starting position. The ‘almost’ feasible fitness values, we argue, are the dead-ends of sub-optimal *massif centrals* or *funnels*.

The success of the MS tells us that using all of: recombination, random mutation, and guided local search together works in harmony with this configuration space to guide the search to promising feasible regions. The genetic algorithm’s vast range of obtained fitnesses suggests a lack of reliability. Sometimes a feasible solution may be found (a previous paper found that it was around 5% of runs given 100 generations [14]) but other times a population filled with individuals of severely penalised fitness may be obtained. Contrarily, the MA appears to be rather uniformly consistent: all of the total 400 runs ended with a feasible fitness, and this was also always ≥ 1.15.

The consistency of the ILS lies somewhere between the performances of the GA and the MS: although often the end fitness is infeasible, the range of values in the distributions is tight compared to the GA and are usually between -10 and +1.66. ILS solutions could be seeded to a different highly-exploitative algorithm to finish the job.

6 Conclusions

This work has pursued modelling a problem from healthcare with LONs. Doing so brought the added complications of infeasible regions in the fitness landscapes, which is new for LON analysis. Two algorithms were offered for the purpose of constructing LONs for CSOP: *Markov-Chain* LON Construction and *Hybrid* LON Construction. An examination of the feasibility gradients within the LONs revealed that an ILS framework (i.e. the *Markov-Chain* algorithm) may be better at finding portals out of infeasible regions but lack in local optima-level exploitative power when in a promising region. MS (i.e. the *Hybrid* algorithm) appears very proficient in exploitation within feasible areas but did not boast many escapes from infeasible areas. This provides insight into how CSOP interacts with sequences of search operators. We showed that our MS and our ILS outperformed the GA from the literature, even when mandating an equal fitness function budget for the algorithms. We stipulate that the selection process of the GA does not have sufficient exploitative power (perhaps due to the small population size) and this can be brought by adding local search. Importantly, the best results are obtained using the recombination and random mutation of the GA *together with* a guided local search. It follows that the former bring innovation and diversification, while the latter brings intensification and facilitates propagation of good genes.

References

1. Ochoa, G., Tomassini, M., Vérel, S., Darabos, C.: A study of nk landscapes' basins and local optima networks. In: Proceedings of the 10th annual conference on Genetic and evolutionary computation. pp. 555–562. ACM (2008)
2. Verel, S., Ochoa, G., Tomassini, M.: The connectivity of nk landscapes' basins: A network analysis. arXiv preprint arXiv:0810.3492 (2008)
3. Herrmann, S., Ochoa, G., Rothlauf, F.: Communities of local optima as funnels in fitness landscapes. In: Proceedings of the 2016 on Genetic and Evolutionary Computation Conference. pp. 325–331. ACM (2016)
4. Daolio, F., Tomassini, M., Vérel, S., Ochoa, G.: Communities of minima in local optima networks of combinatorial spaces. Physica A: Statistical Mechanics and its Applications 390(9), 1684–1694 (2011)
5. Iclanzan, D., Daolio, F., Tomassini, M.: Data-driven local optima network characterization of qaplib instances. In: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation. pp. 453–460. ACM (2014)
6. Verel, S., Daolio, F., Ochoa, G., Tomassini, M.: Sampling local optima networks of large combinatorial search spaces: The qap case. In: Auger, A., Fonseca, C.M., Lourenço, N., Machado, P., Paquete, L., Whitley, D. (eds.) Parallel Problem Solving from Nature – PPSN XV. pp. 257–268. Springer International Publishing (2018)
7. Ochoa, G., Veerapen, N., Whitley, D., Burke, E.K.: The multi-funnel structure of tsp fitness landscapes: a visual exploration. In: International Conference on Artificial Evolution (Evolution Artificielle). pp. 1–13. Springer (2015)
8. Veerapen, N., Ochoa, G., Tinós, R., Whitley, D.: Tunnelling crossover networks for the asymmetric tsp. In: International Conference on Parallel Problem Solving from Nature. pp. 994–1003. Springer (2016)
9. Ochoa, G., Veerapen, N.: Mapping the global structure of tsp fitness landscapes. Journal of Heuristics pp. 1–30 (2017)
10. Simoncini, D., Barbe, S., Schiex, T., Verel, S.: Fitness landscape analysis around the optimum in computational protein design. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 355–362. ACM (2018)

11. Mostert, W., Malan, K.M., Ochoa, G., Engelbrecht, A.P.: Insights into the feature selection problem using local optima networks. In: European Conference on Evolutionary Computation in Combinatorial Optimization (Part of EvoStar). pp. 147–162. Springer (2019)
12. McCall, J., Petrovski, A.: A decision support system for cancer chemotherapy using genetic algorithms. In: Proceedings of the international conference on computational intelligence for modeling, control and automation. pp. 65–70 (1999)
13. Petrovski, A.: An application of genetic algorithms to chemotherapy treatment. (1998)
14. Petrovski, A., Brownlee, A., McCall, J.: Statistical optimisation and tuning of ga factors. In: 2005 IEEE Congress on Evolutionary Computation. vol. 1, pp. 758–764. IEEE (2005)
15. Petrovski, A., Shakya, S., McCall, J.: Optimising cancer chemotherapy using an estimation of distribution algorithm and genetic algorithms. In: Proceedings of the 8th annual conference on Genetic and evolutionary computation. pp. 413–418. ACM (2006)
16. McCall, J., Petrovski, A., Shakya, S.: Evolutionary algorithms for cancer chemotherapy optimization. Computational Intelligence in Bioinformatics 7, 265 (2007)
17. Brownlee, A.E., Pelikan, M., McCall, J.A., Petrovski, A.: An application of a multivariate estimation of distribution algorithm to cancer chemotherapy. In: Proceedings of the 10th annual conference on Genetic and evolutionary computation. pp. 463–464. ACM (2008)
18. Wheldon, T.E.: Mathematical models in cancer research. Taylor & Francis (1988)
19. Tse, S.M., Liang, Y., Leung, K.S., Lee, K.H., Mok, T.S.K.: A memetic algorithm for multiple-drug cancer chemotherapy schedule optimization. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 37(1), 84–91 (2007)
20. Barbour, R., Corne, D., McCall, J.: Accelerated optimisation of chemotherapy dose schedules using fitness inheritance. In: IEEE Congress on Evolutionary Computation. pp. 1–8. IEEE (2010)
21. Petrovski, A., Wilson, A., Mccall, J., et al.: Statistical identification and optimisation of significant ga factors. In: Proceedings of the 5th Joint Conference on Information Sciences. Atlantic City, USA. vol. 1, pp. 1027–1030 (2000)
22. Petrovski, A., McCall, J.: Multi-objective optimisation of cancer chemotherapy using evolutionary algorithms. In: International Conference on Evolutionary Multi-Criterion Optimization. pp. 531–545. Springer (2001)
23. Villasana, M., Ochoa, G.: Heuristic design of cancer chemotherapies. IEEE Transactions on Evolutionary Computation 8(6), 513–521 (2004)
24. Agur, Z., Hassin, R., Levy, S.: Optimizing chemotherapy scheduling using local search heuristics. Operations research 54(5), 829–846 (2006)
25. Ochoa, G., Villasana, M., Burke, E.K.: An evolutionary approach to cancer chemotherapy scheduling. Genetic Programming and Evolvable Machines 8(4), 301–318 (2007)
26. Ochoa, G., Veerapen, N.: Deconstructing the big valley search space hypothesis. In: European Conference on Evolutionary Computation in Combinatorial Optimization. pp. 58–73. Springer (2016)
27. Ochoa, G., Herrmann, S.: Perturbation strength and the global structure of qap fitness landscapes. In: 15th International Conference, Coimbra, Portugal, September 8–12, 2018, Proceedings, Part II. pp. 245–256 (01 2018)