

---

# An Analysis of the Fitness Landscape of Travelling Salesman Problem

**Mohammad-H. Tayarani-N.**

mhtn1g09@ecs.soton.ac.uk

Department of Electrical and Computer Science, University of Southampton, Southampton, UK.

**Adam Prügel-Bennett**

apb@ecs.soton.ac.uk

Department of Electrical and Computer Science, University of Southampton, Southampton, UK.

---

## Abstract

The fitness landscape of the Travelling Salesman Problem (TSP) is investigated for eleven different types of the problem. The types differ in how the distances between cities are generated. Many different properties of the landscape are studied. The properties chosen are all potentially relevant to choosing an appropriate search algorithm. The analysis includes a scaling study of the time to reach a local optimum, the number of local optima, the expected probability of reaching a local optimum as a function of its fitness, the expected fitness found by local search and the best fitness, the probability of reaching a global optimum, the distance between the local optima and the global optimum, the expected fitness as a function of the distance from an optimum, their basins of attraction and a Principal Component Analysis of the local optima. The Principal Component Analysis shows the correlation of the local optima in the component space. We show how the properties of the principal components of the local optima changes from one problem type to another.

## Keywords

Travelling Salesman Problem, fitness landscape, scaling analysis, long-range correlation.

## 1 Introduction

The travelling salesman problem or TSP is probably the most famous and best studied combinatorial optimisation problem. In this paper we describe a detailed analysis of the properties of the fitness landscape for 11 different types of problem ranging from random problems to those motivated by real world constraints. The problems are artificial in that they are all generated by a instance generator, however, many of them share the constraints that would be expected to arise in real problems. For example, we consider instances which could plausibly occur in a drilling problem, or instances arising from a scheduling problem.

This paper follows two previous papers where we investigated the fitness landscape properties of the Maximum Satisfiability (MAXSAT) problem Prügel-Bennett and Tayarani-N. (2011) and graph-colouring Tayarani and Prügel Bennett (2014). This current paper expands on those previous papers, not only in looking at a different problem, but in studying how properties vary over different types of problems. This is intended to be a self-contained paper and not a comparison with MAXSAT or graph-colouring (we intend to do a comparison elsewhere), however, the previous papers informed the present study about what properties are interesting.

“The relation between the Fourier expansion of a landscape and the computational complexity of the optimization problem on the landscape is of great importance for devising practical optimization heuristics. Disappointingly, there seems to be no simple relationship” Stadler

(1996). We have seen little evidence that it is possible to characterise the complexity of a problem in just a few measures. It is our view that combinatorial optimisation is a tough field requiring an understanding of many different aspects of a problem. This paper examines a large number of properties as it is our belief that all these properties could be relevant in the design and choice of heuristic search algorithms. Alas this does not lead to short punchy papers, but forces us to perform large empirical studies. In consequence, this is a long paper, but it provides an illuminating insight into the structure of the fitness landscape, and shows which properties are universal and which depend critically on the details of the instances.

In this paper, we introduce a new method of analysing the fitness landscape. That is, we use Principle Component Analysis (PCA) to study the structure of the local optima. The motivation for this is that many different local optima often have parts of a tour in common. Thus, the set of local optima do not necessarily span the whole search space, but is confined to a low dimensional sub-space. PCA extracts this information. To the best of our knowledge, this is the first time PCA has been used to study the structure of the fitness landscape in this way.

### 1.1 Previous Landscape Analysis

The analysis of fitness landscape goes back to the seminal work of Sewall Wright Wright (1932) and has attracted considerable attention. Examples of the measures that have attempted to capture the ruggedness of the landscape include the Autocorrelation Weinberger (1990); Angel and Zissimopoulos (1998) and Fitness Distance Correlation Jones and Forrest (1995); Merz and B.Freisleben (2000). But it was soon realised that these measures do not necessarily capture the hardness of the problems, when a counter example was proposed which was an easy problem but showing no relationship between the fitness of the solutions and their distance to the global optimum Altenberg (1997). The question of “what makes a fitness landscape hard” continues Forrest and Mitchell (1993); Jones and Forrest (1995), and new methods of capturing the hardness of the problems are proposed, like algebraic properties of the solutions in the landscape Grover (1992); Stadler (1995), modality (number of local optima) Horn and Goldberg (1995) and the fractal dimension of the fitness landscape Hoshino et al. (1998). Some researchers try to explain when a problem becomes hard, by studying the area in the landscape called “Olympus”, in which the better local optima are located Verel et al. (2007); Vérel et al. (2008). Fitness cloud is another method proposed to visualise the fitness landscape which tries to represent some properties of the fitness landscape Collard et al. (2007); Lu et al. (2011); Vanneschi et al. (2007), reflecting the problem hardness. Problems can be difficult for many different reasons. The aim of this paper and its sister papers Prügel-Bennett and Tayarani-N. (2011); Tayarani and Prügel Bennett (2014) is to understand why real world problems, exemplified by some of the classic combinatorial optimisation problems become hard. We have not sought a single measure of hardest (as we believe there can be many reasons for problems to be hard), but rather attempted to provide a full picture of the fitness landscape.

Explaining why and how the problems become hard has not been the only field of research in landscape analysis. Some researchers use the landscape analysis to study some parameters of the Evolutionary Algorithms, like the population size Alander (1999), or operators like mutation and crossover Mathias and Whitley (1992); Suzuki and Iwasa (1997), the recombination operators Hornby (1996), or the perturbation operator Martin et al. (1999). Another field of study has been to study the role of the representation of the solutions, on the landscape properties, and finding the best representation for each problem Tavares et al. (2006); Riley and Ciesielski (2010). Some researchers have used landscape analysis to explain why some algorithms, like local search algorithms Fonlupt et al. (1997), memetic Algorithms (i.e. hybrid genetic algorithms) Merz (2004) or metaheuristic algorithms based on local search Watson (2010), work better on particular landscapes. Some other works have tried to categorise the problems based

on their landscape properties Merz and B.Freisleben (2000).

There are several papers that try to exploit the concept of the landscape and landscape analysis in developing new sets of algorithms for different problems. The fitness landscape analysis is used to propose Memetic Algorithms for Graph Bi-Partitioning problem Merz and Freisleben (1998), Resource Allocation problem Huang et al. (2009) and Maximum Satisfiability problem Zhang et al. (2003); Zhang (2004), or improving the performance of evolutionary algorithms by a landscape approximation Ratle (1998); Pošík and Franc (2007); Shen and He (2010). In a more recent work, the landscape analysis is used to propose a new population based algorithm Qasem and Prügel-Bennett (2010). The landscape of many other problems have been studied, including the landscape of graph drawing Lehn and Kuntz (2001), graph-colouring Hertz et al. (1994); Bouziri et al. (2011); Tayarani and Prügel Bennett (2014), antenna design Alander et al. (2002), maximum satisfiability Prügel-Bennett and Tayarani-N. (2011), flow-shop scheduling Czogalla and Fink (2011) and Bayesian network structure Wu et al. (2011) problems.

The landscape of Travelling Salesman Problem (TSP) has also been studied in a number of different papers. In the first major effort in understanding the landscape of TSP, a big valley structure was observed in the landscape Stadler and Schnabl (1992). Similar big valley structure was subsequently observed in other TSP problem instances Boese (1995), the graph-partitioning problem Wiles and Tonkes (2006) and the graph bijection problem Boese et al. (1994). The most recent work on the big valley hypothesis studies different problems with different sizes using the Concord algorithm DR Hains and Howe (2011). Our paper expands on this previous work in two directions. Firstly, we examine considerably more properties and particularly those related to the structure of the local minima. Secondly, we study how these properties differ depending on the structure of the distance matrix.

The rest of this paper is organised as follows. In the next section we introduce the travelling salesman problem and describe the local search algorithm we use to find the optima in the landscape. Section 3 describes some statistical properties of random solutions, auto-correlation, and some properties of global and local optima including the number of steps a local search algorithm takes to get to a local optimum, number of local and global optima and distance between the optima. We also examine the expected fitness of configurations in Hamming spheres of different radii from a local optimum. We also consider the probability of returning to a local optimum starting from a randomly chosen configuration in the Hamming sphere. The principal component analysis of the local optima is presented in 4 and we draw conclusions in section 6.

## 2 Travelling Salesman Problem

In the Travelling Salesman Problem (TSP) you are given a list of cities and the distances between every pair of them. The task is to find the shortest tour that visits each city for exactly once and then returns to the start. Despite its deceptively simple descriptive it is surprisingly difficult to solve. In general it is NP-hard, although for some problem types such as Euclidean TSP there is a polynomial time approximation scheme Arora (1998). In this paper, we investigate 11 different types of instances, many inspired by real world problems. The categories we have chosen have been studied in other papers Cirasella et al. (2001); Rardin et al. (1993). The problems differ in how the distance matrix is generated.

### 2.1 Problem Types

There are different ways of representing the TSP problem mathematically. One method is in terms of finding a bijection from each city to the successor city  $s : \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}$

which minimises the total tour distance

$$c(s) = \sum_{i=1}^n \mathbf{M}_{is(i)} = \sum_{i,j=1}^n \mathbf{M}_{ij} \llbracket j = s(i) \rrbracket$$

where  $\mathbf{M}_{ij}$  is the “distance” between cities  $i$  and  $j$  and  $\llbracket \text{predicate} \rrbracket$  is an indicator function. For a legal tour we require  $s(i) \neq i$ . We consider 11 different problem types, which differ in construction on the distance matrix  $\mathbf{M}$ . The only common assumption is that the distances,  $\mathbf{M}_{ij}$ , are not negative.

### 2.1.1 Random Asymmetric Matrices

In this problem the distance between the cities is chosen randomly,

$$\mathbf{M}_{ij} = \text{Rand}(0, 10^6), \quad \text{for } i, j = 1 \dots n, \quad (1)$$

where  $\text{Rand}(A,B)$  is a uniform random number generator that generates integer deviates,  $\mathbf{M}_{ij}$ , where  $A \leq \mathbf{M}_{ij} < B$ . Although this set of problems does not belong to the real world problems, many of algorithm designer use this type as a challenge to their algorithms. We refer to this set of problems as the **Random** problems.

### 2.1.2 Random Asymmetric Matrices Closed Under Shortest Paths

The previous set of problems lacks the correlation between the distances, making this set an unrealistic problem type. Taking the random problems made through the previous method and closing them under shortest path computation make a more realistic set of problem. A matrix  $\mathbf{M}$  is closed under shortest path if and only if,

$$\forall i, j, k, \mathbf{M}_{ij} \leq \mathbf{M}_{ik} + \mathbf{M}_{kj}. \quad (2)$$

That is, the “distances” satisfy the triangular inequality. Closing a random matrix can easily be performed by applying the following operator on all the values in the matrix,

$$\forall i, j, k, \text{ if } \mathbf{M}_{ij} > \mathbf{M}_{ik} + \mathbf{M}_{kj} \text{ then } \mathbf{M}_{ij} \leftarrow \mathbf{M}_{ik} + \mathbf{M}_{kj} - \text{Rand}(0, 500). \quad (3)$$

This process is applied until all the values in the matrix satisfy the equation 2. We refer to this set of problems as the **Random C** problems.

### 2.1.3 Random Symmetric Matrices

This class of problems is like the Random problem, except the problem matrix is symmetric here,

$$\forall i, j, \mathbf{M}_{ij} = \mathbf{M}_{ji}. \quad (4)$$

We refer to this set of problems as the **Random S** problems.

### 2.1.4 Random Symmetric Matrices Closed Under Shortest Paths

In this set of problems, the matrices are both Closed and Symmetric. We refer to this set of problems as Random SC problems. In this case, our distances between cities form a proper metric distance.

### 2.1.5 Random Two-Dimensional Rectilinear Problems

In this set of problems, the cities are uniformly distributed in a  $10^6$  by  $10^6$  square and the distance between the cities is computed based on the rectilinear metric,

$$\mathbf{M}_{ij} = |x_i - x_j| + |y_i - y_j|, \quad (5)$$

where  $(x_i, y_i)$  shows the coordinates of the  $i$ -th city. We refer to this set of problems as the **Rectilinear** problems.

### 2.1.6 Tilted Drilling Machine Instances with Additive Norm

This set of the problems are motivated by considering the cost of driving an idealised drilling machine. The task is to drill a collection of holes on a tilted surface where the drill is moved by two motors. The first motor moves the drill in the  $x$ -coordinate and the second motor moves it in the  $y$ -coordinate. Because of the drill weight, the second motor needs less energy when moving the drill down, than moving it up. The problem generator places the holes uniformly in a  $10^6$  by  $10^6$  square and has three parameters,  $u_x$  is a coefficient which shows how much energy the first motor needs to move the drill one unit in the  $x$  direction,  $u_y^-$  is a coefficient which shows the energy needed to move the drill one unit down and  $u_y^+$  which is the needed energy when moving upward. So the problem matrix is generated according to

$$\mathbf{M}_{ij} = \begin{cases} u_x|x_i - x_j| + u_y^+(y_j - y_i) & \text{if } y_i \leq y_j \\ u_x|x_i - x_j| + u_y^-(y_i - y_j) & \text{if } y_i > y_j. \end{cases} \quad (6)$$

We consider  $u_x=1$ ,  $u_y^+=2$  and  $u_y^-=0$ . We refer to this set of problems as the **Additive Drilling Problems**.

### 2.1.7 Tilted Drilling Machine Instances with Sup Norm

For many drilling machines, the cost of drilling will depend on the maximum time the drill moves in either the  $x$  and  $y$  directions rather than their sum. For this problem type the holes are placed just like the previous problem but the distances are specified as follows,

$$\mathbf{M}_{ij} = \begin{cases} \max(u_x|x_i - x_j|, u_y^+(y_j - y_i)) & \text{if } y_i \leq y_j \\ \max(u_x|x_i - x_j|, u_y^-(y_i - y_j)) & \text{if } y_i > y_j. \end{cases} \quad (7)$$

For this problem type we consider  $u_x=2$ ,  $u_y^+=4$  and  $u_y^-=1$ . The points are randomly and uniformly put in a  $10^6$  by  $10^6$  square and take integer values. This problem type is referred to as the **Sup-Drilling** problem.

### 2.1.8 Random Euclidean Stacker Crane Problem

We consider a crane having to move a set of  $n$  objects. Each object is at a (source) location  $\mathbf{s}_i$  and has to be moved to a destination location  $\mathbf{d}_i$ . The problem is to choose the order of the tasks to minimise the Euclidean distance between the destination location of one task and the source location of the next location. That is, we take the distance to be

$$\mathbf{M}_{ij} = \|\mathbf{s}_i + \mathbf{d}_i\| + \|\mathbf{d}_i - \mathbf{s}_j\|,$$

where  $\|\cdot\|$  denotes the Euclidean distance. The sources are uniformly picked up from a  $10^6$  by  $10^6$  square,

$$\mathbf{s}_i = \left( \text{Rand}(0, 10^6), \text{Rand}(0, 10^6) \right), \text{ for } i = 1 \dots n, \quad (8)$$

while the destinations are chosen to be “close” to the sources,

$$\mathbf{d}_i = \mathbf{s}_i + \left( \text{Rand}\left(0, \frac{10^6}{\sqrt{n}}\right), \text{Rand}\left(0, \frac{10^6}{\sqrt{n}}\right) \right), \quad (9)$$

where  $n$  is the number of source-destination pairs (problem size). We denote this problem by **crane**. In figure 1 we show an example of a possible tour made by the crane.

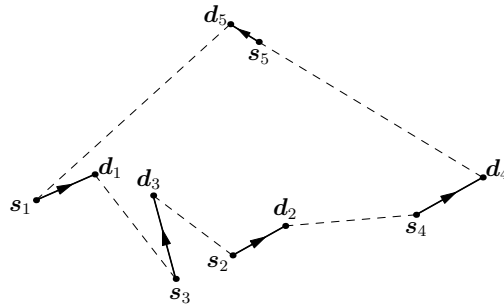


Figure 1: Example of the crane problem for five “cities”. The dashed line show one possible tour around the “cities”.

### 2.1.9 Disk Drive Problem

These instances are generated by an idealised model of the scheduling job of the read head of a computer disk. The task is to extract  $n$  records from a disk. This problem is similar to stacker crane problem in that the files to be read have a start position and an end position in their tracks. The position is represented in polar coordinates, with the first coordinate representing the radius and the second coordinate the angle. The destination is at the same radius as the source. This is illustrated in figure 2, which shows an example of five records stored on a disk drive. The read head has to move from the destination of the current record to the source of the next record. We assume that changing the position of the read head (it radius) is 10 times slower than disk rotation.

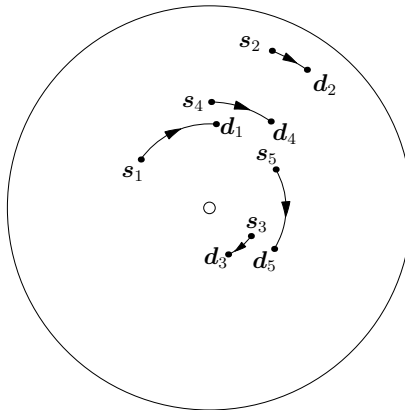


Figure 2: Example of the disk problem with five records.

The source points are generated using equation 8 while the corresponding destination is generated using

$$d_i = (s_{i1}, s_{i2} + \text{Rand}(0, 10^5) \bmod 10^6),$$

where  $\mathbf{s}_i = (s_{i1}, s_{i2})$  is the coordinate of the source. This models the situation where the record lies on the same track as the source and the disk has circular boundary conditions in the first coordinate. Furthermore, we assume that the head is moved in the second coordinate, but must wait until the spinning disk reaches the correct first coordinate. The head motion is assumed to

be 10 times slower than the disk motion. The cost is taken to be the time to access all the records and return to the starting position as follows,

$$M_{ij} = 10|d_{i1} - s_{j1}| + |d_{i2} - s_{j2}|.$$

Although, this is clearly idealised (for example, no account is made of acceleration and deceleration times of the read head), nevertheless, it captures much of the structure that a real disk problem would have. We refer to this problem as **Disk Drive**.

### 2.1.10 Euclidean Problem

This is the well-known Euclidean TSP, the description of which can be found in Cirasella et al. (2001). To generate instances we uniformly and randomly select the cities in a  $10^6 \times 10^6$  square, and the distance between the cities is the Euclidean distance between them. We refer to this problem type as **Euclidean**.

### 2.1.11 No-Wait Job-Scheduling

In this problem we assume that we have  $k$  processors and a set of  $n$  tasks, each of which must use the  $k$  processors in order with no waiting time allowed between ending one task and starting the next. The no waiting condition would be realistic, if, for example, the processing required the material to be kept hot, or alternatively, if there was no storage space between two consecutive tasks. We denote, the times taken to complete the  $l^{th}$  process for the  $i^{th}$  task as  $u_l^i$ . Thus each task can be represented by a list of sub-task times  $\mathbf{u}^i = (u_1^i, u_2^i, \dots, u_k^i)$ . What makes the scheduling problem non-trivial is that you can start the next task before the previous task is finished, provide that none of the sub-tasks have to wait. Thus the distance between tasks is the wait time

$$M_{ij} = \max_{l \in \{1, 2, \dots, k\}} \left( \sum_{r=1}^l u_r^i - \sum_{s=1}^{l-1} u_s^j \right),$$

We illustrate this wait time for two problems in figure 3. In this paper we consider  $k = 5$  and the times  $u_l^i$  are made by a uniform random number generator between 1 and 1000.

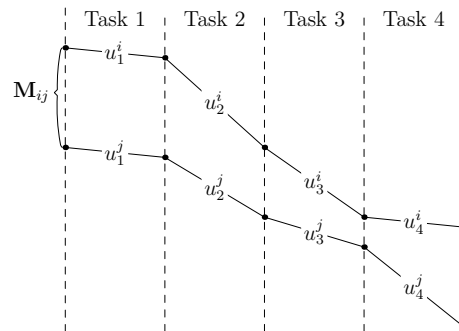


Figure 3: Example of the wait time between process  $\mathbf{u}^i$  and  $\mathbf{u}^j$  for a problem consisting of 4 tasks. If a smaller waiting time,  $M_{ij}$  were used then task 3 in process  $j$  would have to wait for processor 3 to complete task 3 for process  $i$ , but this is not allowed.

## 2.2 Local Search Operator

To define the landscape of a problem requires some definition of the neighbourhood of the configurations. The neighbourhood is usually defined in terms of the configurations that can be

reached by some local search operator. We consider  $k$ -opt moves which are the most commonly used move set used by heuristic algorithms to solve TSP. A  $k$ -opt move consists of dividing the tour into  $k$  segments and then recombining the segments in such a way as to obtain a legal tour. The simplest  $k$ -opt move is 2-opt. For most of the analysis we carried out we found 2-opt impractical for two reasons. Firstly, for tours of moderate size ( $n$  around 50) we found that the number of local optima was so large that we were unable to store them. Secondly, the 2-opt move reverses one segment of the tour. For asymmetric problems this would disrupt a large number of edge distances so 2-opt cannot be viewed as a minimal move for this set of problems. Thus, in this paper, we focus on 3-opt moves, which both substantially reduces the number of local optima and contain moves which change the tour, but does not necessarily change the direction in which a segment is traversed. We also considered using 4-opt, however, we found that this had such a large neighbourhood it became computationally very expensive to ensure that a local optimum had been reached.

Note that there is a difference in the interpretation of distance between symmetric and asymmetric problems. In symmetric problems the edges  $(i, j)$  and  $(j, i)$  are treated as the same, while they are treated as different edges for asymmetric problems. If we swap three bonds in a 3-opt this would correspond to making a move of size 3 in Hamming distance for symmetric problems. For asymmetric problems the distance would also depend on whether any of the segments of the tour were reversed. Reversing a segment would on average change around  $n/3$  edges for asymmetric problems.

Although, it is convenient to consider  $k$ -opt neighbourhoods, it is often easier to use the Hamming distance in the space of edges rather than the minimum number of  $k$ -opt moves needed to go from one configuration to another. That is, we can represent a tour as a binary vector in the space of edges, with 1 denoting the existence of an edge in the tour. For symmetric problems there are  $n(n-1)/2$  edges, while for asymmetric problems there are  $n(n-1)$  edges. Each tour consists of exactly  $n$  non-zero components. The Hamming distance counts the number of edges that differ between two tours.

### 2.3 Exhaustive Local Search Algorithm

One way of studying the topology of the local optima is to check all the solutions in the landscape and find every local optimum. This strategy is feasible for the problem sizes up to 20 or 30. However, as the search space for TSP grows as the factorial of the system size, it is impossible to do such an experiment on larger problems. In order to study the local optima for larger problem sizes we use a local search algorithm which exhaustively explores the neighbours of the current solution and any neighbouring solutions at the same cost to find an improving move. A comprehensive description of the exhaustive local search algorithm which guarantees that the solution it returns is located on a local optimum and returns a unique ID for each local optimum can be found in Tayarani and Prügel Bennett (2014). The algorithm starts from a random configuration and uses the 3-opt local search process to find a better solution until a local optimum is reached. It then returns a unique ID for the local optimum it has found.

## 3 Landscape Analysis

This section explores many properties of the landscape of TSP and in particular it focuses on how these properties scale as the problem size increases.

### 3.1 Density of States

We start our analysis by studying the statistical properties of randomly drawn solutions in the landscape. In our experience these properties do not correlate well with the problem hardness, but it provides information on the probability of finding an improving neighbour at a given



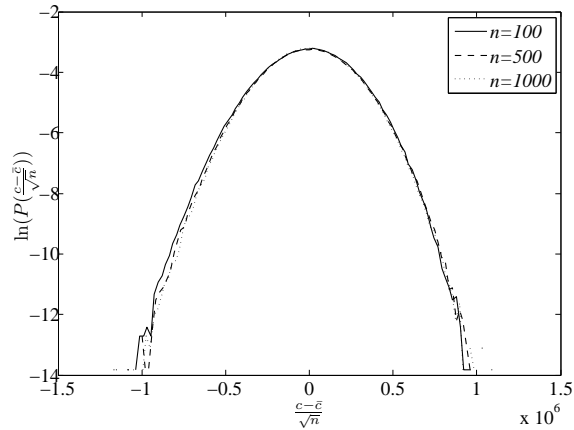


Figure 4: Natural Logarithm of histogram of costs for random solutions of particular instances of Euclidean problem for  $n = 100, 1000, 10000$ .

cost level. This might provide useful information, for example, in deciding the size of the neighbourhood in local search.

The density of states shows the number of configurations at each cost level. The average cost of the solutions in TSP is the average distance between the nodes. By randomly sampling the solutions and finding the histogram of cost of the random solutions, we can compute the spread of costs around the mean. Figure 4 shows the natural logarithm of the histogram of costs around the average cost scaled by  $\sqrt{n}$  for  $n = 100, 500$  and  $1000$  for the Euclidean problem, where  $n$  is the size of the problem. The results are similar for every randomly drawn problem instance. The cost of a random solution is approximately normally distributed around the average cost with a variance that grows approximately as  $n$ . We did the same experiment on all the 11 problem types. In each case we averaged over 50 different problem instances of each type. We observed very similar behaviour for all the problem types.

### 3.1.1 Statistical Measures

The density of states for all problems is a bell-shaped curve resembling a normal distribution. To quantify the difference between the problems, we can measure the cumulants,  $\kappa_n$ . The first cumulant is the mean, the second cumulant is the variance, while the higher cumulants provide a natural measure of the deviation from the mean. The third and fourth cumulant can also be expressed in terms of the central moments

$$\mu_n = \mathbb{E}((c - \mathbb{E}(c))^n),$$

where  $c$  is the cost of the tour and the expectation is over all instances of the problem. The third cumulant is equal to the third central moment  $\mu_3$  while the fourth cumulant is equal to the fourth central moment minus three, times the variance squared. In a normal distribution the third and fourth cumulants are zero. The higher-order cumulants are invariant to changing the definition of the costs by an additive constant, but are not invariant to the scale of the cost. A more useful set of measures to understand how much a distribution differs from a normal distribution is to consider the scaled cumulants where we scale the  $n^{\text{th}}$  cumulant by a factor of  $\kappa_2^{n/2}$ . The scaled cumulants are now invariant to an affine transformation of the costs  $c \rightarrow ac + b$  (for  $a > 0$ ). The third scaled cumulant is known as skewness, denoted by  $\gamma_1$ , while the scaled fourth cumulant is known as the kurtosis and is denoted by  $\gamma_2$ .

In principle we could compute the cumulants *ab initio* (we did this for graph-colouring in the previous paper in this series Tayarani and Prügel Bennett (2014)). For example, the mean is equal to

$$\kappa_1 = \mathbb{E} \left( \sum_{i,j=1}^n \mathbf{M}_{ij} \llbracket j = s(i) \rrbracket \right).$$

For random tours  $\mathbb{E}(\llbracket j = s(i) \rrbracket) = \llbracket j \neq i \rrbracket / (n - 1)$  so that

$$\kappa_1 = \frac{1}{n-1} \sum_{i,j=1}^n \mathbf{M}_{ij} = n \bar{\mathbf{M}}$$

where  $\bar{\mathbf{M}}$  is the mean of the distance values. We can thus compute the mean length of a random tour averaged over all instances by computing the average distance for the 11 problems. For example, for a 2-D Euclidean tour with cities randomly placed in a unit square

$$\begin{aligned} \bar{\mathbf{M}} &= \int_0^1 \int_0^1 \int_0^1 \int_0^1 \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2} dx_0 dx_1 dy_0 dy_1 \\ &= \frac{1}{15} \left( 2 + \sqrt{2} + 5 \sinh^{-1}(1) \right) \approx 0.521. \end{aligned}$$

Such calculations, however, are not very informative. Furthermore, for higher cumulants they rapidly become very complicated, we therefore do not pursue these calculations, but rather show empirically obtained results.

In TSP, as the size of the problem grows, the skewness and kurtosis of the cost of the random solutions change. Figure 5 shows the relationship between the skewness and kurtosis of the distribution of the costs for the Euclidean problem for different problem sizes. For small problems the skewness and kurtosis are negative. The negative skewness means that for small problems the bulk of the costs lies to the right of the mean and the negative kurtosis means that costs are more distributed around the mean than a normal distribution. The only problem type which has a positive skewness is the Random problem class. For Random Symmetric problems the skewness is around zero for all the problem sizes.

The skewness versus the kurtosis of the distribution of costs for different problem types for  $n=10$  and  $n=1000$  are represented in figure 6. The results are averaged over 50 different problem instances and  $10^6$  sampling for each. It is clear that for all the problem types as the system size increases, the distribution of the costs converges towards a normal distribution. (This might seem an inevitable consequence of the central limit theorem, however it is not true, for example, in graph colouring the skewness and kurtosis remain non-zero as  $n \rightarrow \infty$  Tayarani and Prügel Bennett (2014).)

### 3.2 Auto-Correlation

One way of measuring the ruggedness of the fitness landscape is to find the auto-correlation of a random walk through the search space Weinberger (1990). In order to compute this we use a random walk algorithm, starting from a random point and moving to a neighbour at each step. The auto-correlation of the landscape is given by,

$$R(\tau) = \frac{1}{\sigma^2} \mathbb{E} \left( (c(t + \tau) - c_{av})(c(t) - c_{av}) \right), \quad (10)$$

where  $c(t)$  is the cost at step  $t$  and  $\sigma^2$  is the variance in the cost for random solutions. We have computed the auto-correlation for different sizes of different types of the problem. The curves

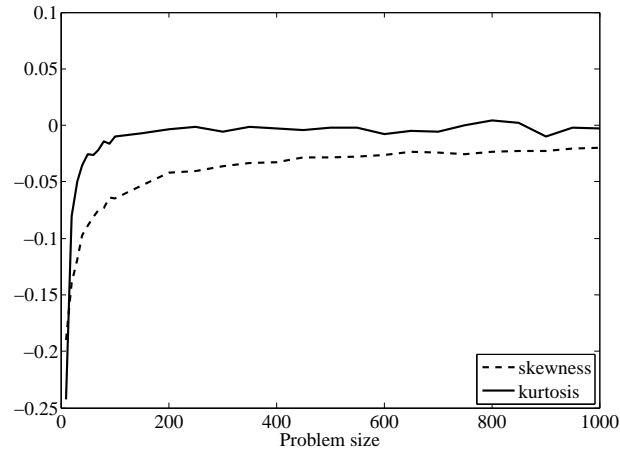


Figure 5: The skewness and kurtosis of the distribution of costs for Euclidean problem instances.

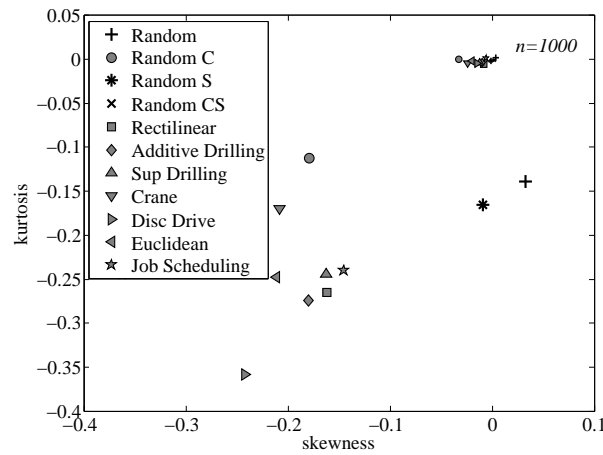


Figure 6: The kurtosis of randomly constructed solutions versus skewness for different problem types with different sizes. The small symbols represent the problems with  $n=1000$  and the large symbols represents the problems with  $n=10$ . The results are computed by sampling  $10^6$  random configurations for 50 different problem instances.

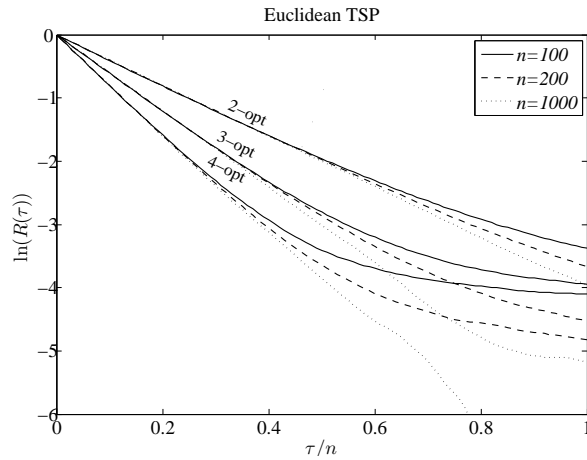


Figure 7: Natural logarithm of auto-correlation for three instances of size  $n=100$ , 200 and 1000, for Euclidean problem, plotted against the time difference  $\tau/n$ . The number of steps is  $10^8$ .

representing the auto-correlation for particular instances of Euclidean problem for  $n=100$ , 200 and 1000 for three different walks, 2-opt, 3-opt and 4-opt are shown in figure 7. Note that for large values of  $\tau$  the estimated value of  $R(\tau)$  is unreliable due to statistical fluctuations, even though we used  $10^8$  steps in our estimate. Under the rescaling used in this figure, the graphs for all the problems are very similar. The auto-correlation function appears to fall off approximately exponentially as,

$$R(\tau) \approx e^{-\tau/l},$$

where  $l$  is known as the correlation length Stadler (1996). The gradient of the curves for 2-opt in figure 7 is about -1.94, therefore for 2-opt  $l \approx 0.51 \times n$ . The correlation length is considered as a measure of the landscape ruggedness—the smaller  $l$  the more rugged the landscape. As  $n$  increases the ruggedness decreases, that is, if we consider two instances of size  $n$  and  $n'$  we get roughly the same degree of ruggedness if we make  $n'/n$  random steps on the instance of size  $n'$  as we would for a single step on the instance of size  $n$ . The correlation length shows that the landscape of TSP is relatively smooth with long range correlation. The landscape correlation will be discussed later. It is also clear that for  $k > k'$ , the correlation between the solutions in  $k$ -opt is weaker than that in  $k'$ -opt. This is because as we increase  $k$  we are making larger steps and thus decorrelating faster.

Figure 8 shows the auto-correlation function for all 11 problem types for  $n=1000$  where we use a 3-opt walk. We observe that the problems split into two major groups. For Random Symmetric, Random Symmetric Closed, Rectilinear, Additive-Drilling, Crane, Disc Drive and Euclidean problems the auto-correlation is a straight line. While for The Random, Random Closed, Sup-Drilling and Job Scheduling the curves have a distinct kink at the beginning. The kink appears at  $\tau/n = 0.02$ .

The kink occurs in those problems with asymmetric weights. The explanation of the kink is that typically a random 3-opt move will reverse the direction in which a tour is visited. For asymmetric problems this causes a rapid decrease in the correlation. However, later moves can reverse the direction that these cities are visited to the original direction, thus restoring the correlation.

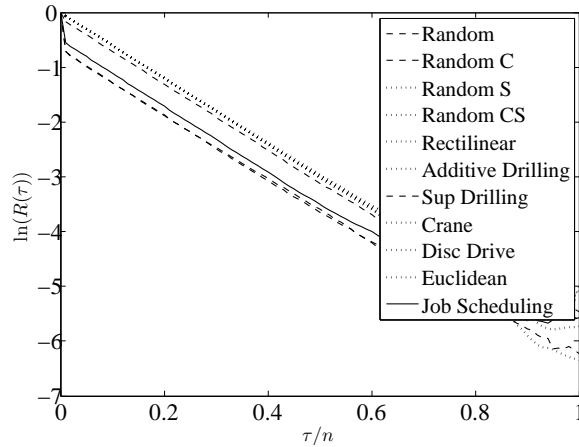


Figure 8: Natural logarithm of the auto-correlation for different problem types with the size of 1000, plotted against the time difference  $\tau/n$ . The number of steps is  $10^8$ . This is for 3-opt operator.

We can measure the degree of asymmetry of a problem using

$$A(G) = \frac{\sum_{i=1}^n \sum_{j=i+1}^n |G_{ij} - G_{ji}|}{\sum_{i=1}^n \sum_{j=1}^n G_{ij}}. \quad (11)$$

In figure 9 we show  $\ln(R(1))$  versus the degree of asymmetry. We note that there are some problems, notably the Additive-Drilling problem and the Sup-Drilling problem, which have a reasonably high asymmetry, but do not show a marked decrease in the correlation produced by a single step. The reason for this is that they contain a hidden symmetry. If you reverse a segment of a tour for these problems the cost depends only on the difference in height of the starting and end point of the segment. As a consequence, these tours are much closer to the symmetric tours than the degree of asymmetry would indicate.

### 3.3 Time to Local Optimum

In the study of the landscape of TSP, we will mainly focus on the properties of local and global optima. Our analysis begins by the study of the number of steps taken by the local search algorithm to reach a local optimum. We count the number of 3-opts performed on the starting configuration, until reaching a local optimum. Note that for symmetric problems, the 3-opt operator changes three edges at each step, and for asymmetric problems many of the edges may change. To check if the local search algorithm has reached a local optimum, an exhaustive local search algorithm is used to make sure that the local search algorithm is at a local optimum.

Figure 10 shows the mean time to reach a local optimum versus  $n$  for all the problem types. It is clear that for all the problem types the time to local optima increases linearly with the system size (with the possible exception of Job Scheduling, which seems to grow slightly faster). The data suggest that the time to local optima increases as approximately  $0.78n$ . From the point of view of time to local optima, the problem becomes linearly harder as the system size grows. But of course it is not only the time that matters; the number of local optima is another important property which determines how hard the problem is. We will discuss this in the next sections.

Although the average time to reach a local optimum is quite similar for different problem types, the distribution of run times differs markedly between problem types. This is illustrated

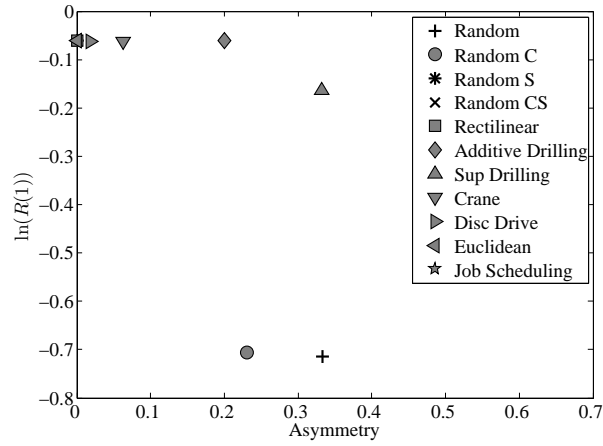


Figure 9: The correlation length for the first steps for different problem types against the asymmetry of the matrix. The size of the problem is  $n=1000$ .

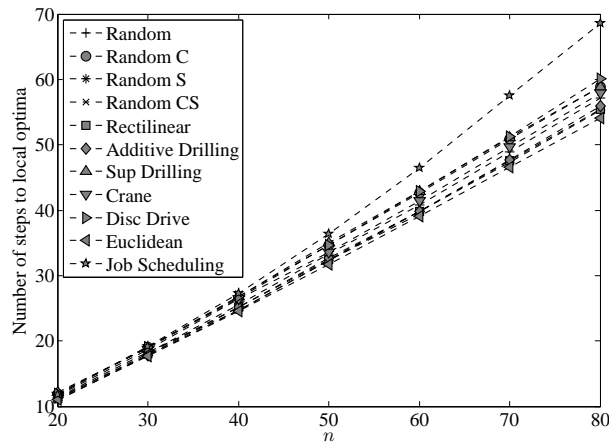


Figure 10: Time to local optima versus system size for different problem sizes.

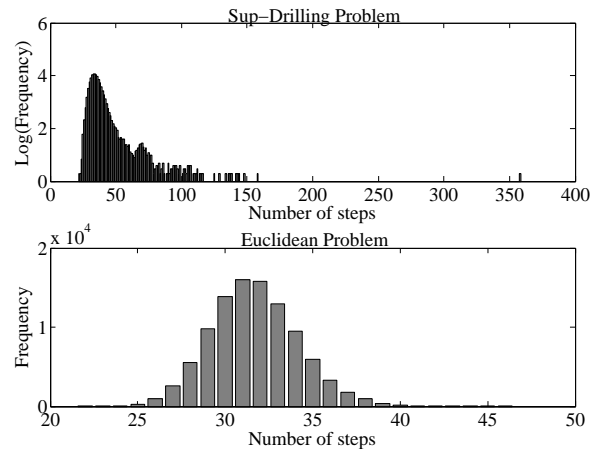


Figure 11: The histogram of number of steps to local optima. The size of the problem is  $n=50$ . The data are for 10 different problem instances and  $10^4$  descents on each of them.

in figure 11, where we show the histogram run times for the Sup-Drilling problem and the Euclidean problem. We have plotted the log-frequency for the Sup-Drilling problem to show the rare events. Although the average number of steps is approximately the same for two different problem types (34.73 for Sup-Drilling and 31.67 for Euclidean), the Sup-Drilling problem shows a long tail at the right side of the mean, while the Euclidean problem shows a more symmetric shape with no long tail. The reason for such difference lies in the values of the distance matrices of the problems. For the Euclidean problem, the distances between the cities are real numbers, so even if truncated, it is exponentially rare to have two edges with the same length. This characteristic makes the fitness of almost all the solutions in the search space different from each other. Having such a characteristic, the landscape tends to have no plateau, so starting from a random solution; the local search algorithm directly reaches a local optimum. For the Sup-Drilling problem, the edges in the graphs take integer values, consequently there can be many different tours of the same length. This produces plateau regions which have to be explored, thus increasing the run time.

So far we have studied the number of steps to reach a local optimum, but we have not shown how the cost reduces during the run. In figure 12 we show the cost versus the number of steps taken averaged over 1000 descents for an instance of the Sup-Drilling problem. We truncate the curves after the optima are reached. We plot the same data using a density plot and on a logarithmic scale in figure 13. The most prominent characteristic is that the improvement in cost decreases roughly exponentially with the number of steps. However, there is clearly, a large deviation from this for a large number of steps caused by the relatively small number of runs that take a long time. These characteristics are shared by all problems, although the exceptionally large tail is a feature only of those problems with large plateaux.

### 3.4 Number of Local Optima

What makes many combinatorial optimisation problem instances difficult is the number of local optima. To find the number of local optima in the landscape we use an exhaustive local search algorithm and store the local optimum it returns. The exhaustive local search algorithm is repeated for a large number of times from randomly chosen starting configurations. Each local optimum with the number of times it is hit is stored. Of course, there is no guarantee that all

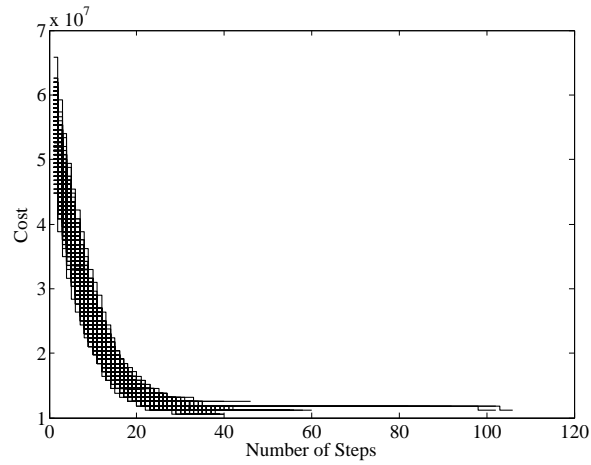


Figure 12: Cost of the solutions the local search algorithm visits at each step, for 1000 descents. This is for Sup-Drilling problem for the size of  $n=50$ .

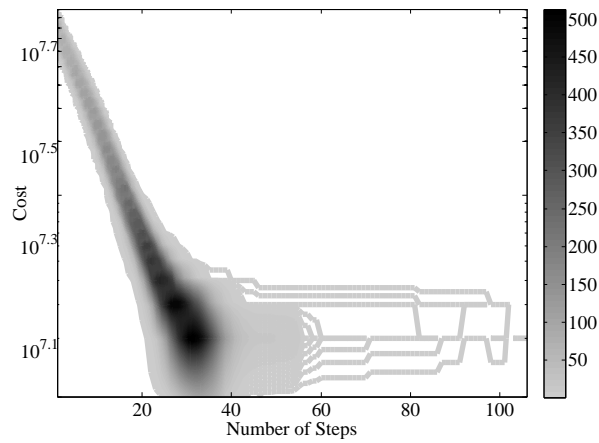


Figure 13: Density graph of logarithm of the cost of the solutions the local search algorithm visits at each step, for 1000 descents. This is for Sup-Drilling problem with the size of  $n=50$ .



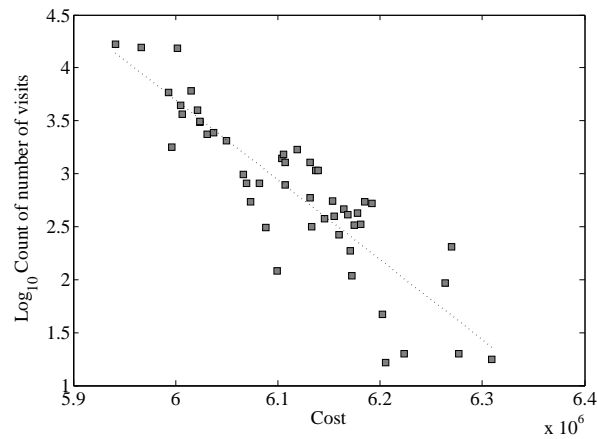


Figure 14: Logarithm of the number times the local search algorithm hits local optima versus the cost of the local optima. The size of the problem is  $n=50$ . This is for one instance of the Euclidean problem. The number of descents is  $10^5$ .

the local optima in the landscape are found, particularly if a local optimum has a small basin of attraction. Typically there are many local optima which have a very small probability of being visited.

The logarithm of the number of times each local optimum is hit versus the cost of the local optimum for the Euclidean problem is shown in figure 14. The data are made based on one single instance and  $10^5$  descents. Of course there are likely to be local optima that we have not found. However, note that local optimum with the highest cost (the worst local optimum) is visited 18 times. The minimum number of times a local optimum is visited is 13. Thus with high probability any local optimum with a cost more than the worst local optimum, that we have not found, would have a basin of attraction 18 times smaller than those that we have found. Although we cannot rule out the existence of such local optima theoretically or practically, we strongly doubt their existence as we have not found any example of them in a very large number of trials.

We observe that gradient of the fitting line in figure 14 is  $-7.29 \times 10^{-6}$  which implies that the probability of ending up in a local optima of cost  $c$  decays exponentially as  $P_v(c) \propto 10^{-0.00000729c}$ . Using the extrapolation we find for the  $10^5$  descents, the expected number of times we would visit a local optima with a cost greater than  $6.6 \times 10^6$  would be less than 1, if any such optima exist. The probability of finding the global optimum was 16.7% while the probability of finding the second best optimum was 14.9%.

Although for all the problem types the probability of getting to a local optimum decays exponentially as the cost of the local optimum grows, the decay rate is different for different problem types. Since the average cost of the local optima is different for different problem types, in order to be able to compare the decay rate of different problems, the cost of the local optima has to be normalized. To do so, the cost on the horizontal axis in figure 14 is divided by the average cost of the local optima. After normalization, the gradient of the fitting line to the data shows the decay rate of the probability of getting to a local optimum as a function of its cost. The probability of getting to the global optimum versus the decay rate for all the problem types is shown in figure 15. For the Random problems we found the best solution on average 1.07 times in  $10^5$  descents. Thus for this problem, we can have no confidence that we have found the global

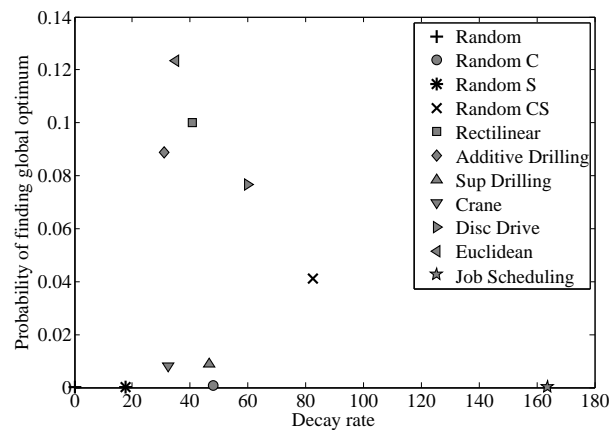


Figure 15: The probability of getting to the global optimum solution versus the decay rate for all the problem types. These are averaged over 10 different instances and  $10^5$  descents on each.

optimum. There seems little evidence of a relationship between the cost of the optima and the size of their basin of attraction. When the Random problems become symmetric, the probability of getting to the global optimum grows; on average the best solution is hit 20 times. When the Random problems are forced to satisfy the triangle inequality (Random Closed) the correlation between the cost of a local optimum and the probability of it being visited increase further. For this problem type, the average number of times the best found optimum is hit is 80 times in  $10^5$  descents. Going from Random problems to Random Closed problems, the decay rate grows as well. When the Random problems are both symmetric and closed, then the number of times the global optimum is hit grows to 4 119 times in  $10^5$  descents. For the case of the Job-Scheduling problem, the best observed solution on average is hit 4 times in  $10^5$  descents. This means that in comparison with the Euclidean problem for example, in which the best observed solution is visited for more than 10 000 times, the best observed local optimum in Job-Scheduling problem does not show any significant superiority over other local optima, in terms of its size of the basin of attraction. It is interesting, as the Job-Scheduling problem is not a random problem, we might expect to observe a correlation between the cost of a local optimum and the size of its basin of attraction, but figure 15 shows no such relation.

Problem types Sup-Drilling and Crane show a quite similar behaviour. For the Sup-Drilling problem the global optimum on average is hit 695 times and in the Crane problem it is hit 853 times in  $10^5$  descents. For the four remaining types the decay rate is similar but the probability of getting to the global optimum is different. The Euclidean problem shows the highest probability of getting to the global optimum. In the case of figure 14 the best observed optimum is hit 16 752 times in  $10^5$  descents.

The histograms of the number of local optima at each cost for two different problem types are shown in figure 16. The histogram shows different behaviour for different problem types; for the Sup-Drilling problem it is close to a normal distribution, while for the Euclidean problem it is closer to a uniform distribution. In some cases like Job-Scheduling, the distribution has a long tail at the right side. A tail at the right side of the distribution means that there could be high cost local optima but with smaller probability of being visited. The distribution for the Euclidean problem shows no tail at either of its sides. There is quite a large proportion of local optima at both high and low cost levels.

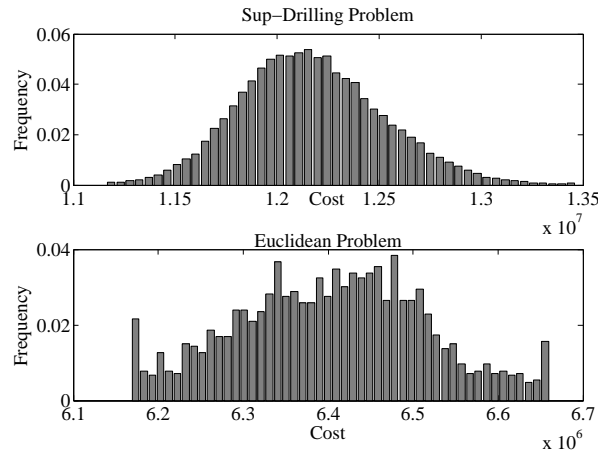


Figure 16: The histogram of the number of local optima at each cost, for two different problem types. The size of the problem is  $n=50$ . The data are for 10 different problem instances and  $10^4$  descents on each of them.

One way of measuring the size of the tails is to consider the distribution of costs of the local optima and measure the proportion of local optima in the 10% of highest and lowest costs. This is shown in figure 17. There is clearly a strong correlation indicating some degree of symmetry. The Euclidean problem has the highest proportion of local optima at the extremes of the cost distribution, while the random problems have the lowest proportion.

### 3.4.1 Growth in the Number of Local Optima

The number of local optima grows with system size. If we take the logarithm of the number of local optima as a function of the cost and scale by a function of the system size we obtain very similar profiles. This is illustrated in figure 18 where we show the natural logarithm of the number of local optima at each cost, divided by  $n \ln(n)$  for the Euclidean problem for different problem sizes. The results are averaged over 20 different problem instances and  $10^4$  descents on each. This suggests that the number of local optima at each cost level grows exponentially as

$$N(c) \approx \exp\left(n \ln(n) g\left(\frac{c - c_{min}}{C_{max} - C_{min}}\right)\right),$$

where  $g(x)$  is the roughly quadratic as shown in figure 18. Figure 14, shows as the cost of a local optimum increases, the probability of the local search algorithm finding it decreases exponentially, so the number of local optima at higher costs (right part of the graph) is underestimated both in figure 16 and figure 18. The plot shows that the histograms are quite similar for different system sizes. Such behaviour is consistent with the hypothesis that the number of local optima grows exponentially with the  $n \ln(n)$ . In order to fit the histogram for different system sizes, in case of the Euclidean problems, the natural logarithm of the number of local optima has been scaled by  $n \ln(n)$  (note that the size of the search space also grows as  $n! = \exp(\Theta(n \log(n)))$ ). Although the number of local optima for all the problem types grows exponentially as a function of the system size, the scaling behaviour changes across different the problem types. For example, for Rectilinear, Disc-Drive and Job-Scheduling problems the scaling is  $n$  and for the Sup-Drilling and Crane problems is  $\sqrt{n}$ . For all the four Random types a vast majority of the local optima are hit once in  $10^5$  descents so it is hard to estimate the relationship between the

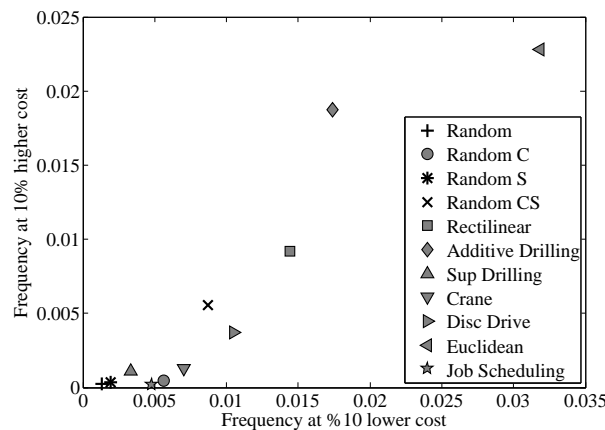


Figure 17: The frequency of the local optima at the higher 10% costs versus the lower 10% costs. The size of the problem in  $n=50$  and the results are computed by sampling  $10^4$  descents for 50 different problem instances.

system size and the number of local optima. Although for all the problem types the number of local optima increases exponentially as some function of the system size.

It is interesting to note that the difficulty of the problem is not obviously correlated with the growth in the number of local optima. For example, Euclidean TSP has a larger probability of finding the global optima than Sup-Drilling despite having significantly greater growth in the number of local optima than Sup-Drilling.

### 3.4.2 Probability of visiting an optimum

We have seen that for some problems the probability of finding a local optimum is correlated with its cost, while this is not true for other problems. We illustrate this in figure 19 which shows the proportion of local optima at each cost level and the probability of finding local optima at the cost level. For the Sup-Drilling problem, the lower cost solutions tend to have significantly larger basins of attraction. In contrast, in the Job-Scheduling problem the sizes of the basins of attraction are almost uncorrelated with the cost.

Clearly, it makes the problems easier to solve when their basins of attraction are larger for lower cost solutions. To measure how the size of the basin of attraction of the local optima changes with the cost we define the bias to be the difference between the expected cost of all the optima and the expected cost of the optima found by local search, divided by the standard deviation in the cost of all the optima. A large bias, indicates that the local search is much more likely to find a fit local optimum than a less fit one. Figure 20 shows the probability of getting to the global optimum versus the bias averaged over 10 different instances for each problem type. There is clearly some correlation between the probability of finding the global optimum and the bias, but interestingly this is not that strong.

### 3.5 Reaching the Global Optima

As we increase the system size the gap between the expected cost found by the local search and the minimum cost increases. This is illustrated in figure 21, where the expected minimum cost (averaged cost of the global optima over 20 problem instances) and the expected cost of the local optima found by the local search algorithm over the same ensemble of randomly drawn instances are shown. We also show the expected cost of the local optima found by local search

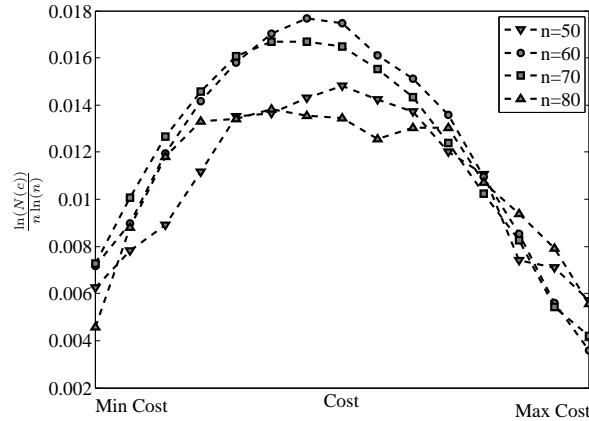


Figure 18: Natural logarithm of the number of local optima at each cost divided by  $n$  for Euclidean problem for different problem sizes. This is averaged over 20 problem instances and  $10^5$  descents on each.

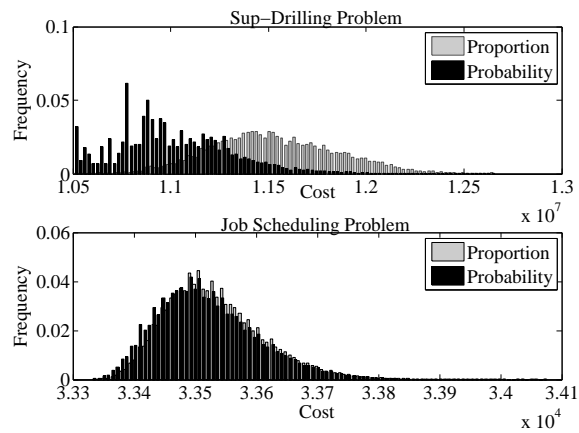


Figure 19: Histograms showing the proportion of local optima at a particular cost and the probability of finding a local optimum at a particular cost one instance.

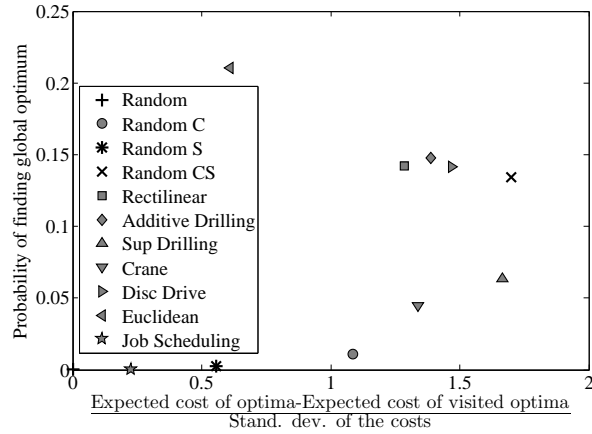


Figure 20: The probability of getting to the global optimum versus average of the cost of the local optima minus the expected cost of the visited local optima divided by the standard deviation of the cost of the local optima for all the problem types. The size of the problem is  $n=50$ . The data are averaged over 10 different problem instances and  $10^5$  descents on each.

plus and minus one standard deviation. It is clear that as the system size grows the gap between the expected cost and the cost of the global optimum solutions grows. For problem sizes  $n$  up to 30, the cost of the global optimum solution lies between the expected cost of the local optima minus one standard deviation. For the size of 80, it lies between the expected cost and expected cost minus two standard deviation. Clearly as the system size grows it becomes less likely to find the global optimum.

Note that in this paper we are not sure if we have found the global optimum. So by global optimum we simply mean the best observed solution. However, as shown above for many of the problem types, the better local optima have exponentially larger basins of attraction. As in our experiments the best found local optimum has been visited for thousands of times, we believe it is highly unlikely to have an optimum with a better fitness than the best observed one and yet a basin of attraction thousands of times smaller. Therefore the best observed solution, with a high probability, is the global optimum (although this is not true for random problems of the Job Scheduling problem).

We can also compute the probability of finding the best found local optimum directly. This is shown in figure 22 where we plot the log-probability of finding the best optimum versus system size for four different problem types. The data are averaged over 20 different problem instances and the number of descents is  $10^4$ . The straight line fit is consistent with the hypothesis that finding global optima becomes exponentially unlikely as the system size grows. Using the straight line fit in figure 22 to extrapolate to large  $n$ , the probability of the local search algorithm finding a global optimum for an instance of size 1000 of the Euclidean problem would be  $6.43 \times 10^{-23}$  and  $1.62 \times 10^{-38}$  for the Sup-Drilling problem. Although such extrapolation is unlikely to provide a precise value, nevertheless, it provides a strong indication that for larger problem instances, using multiple runs of local search algorithms begins to be useless. For random and job scheduling problems the number of times the best visited optimum is visited, rapidly shrinks to one in  $10^4$  descents. For these problem types, for  $n$  larger than 40, the best visited solution is found just once, so we can almost be sure that for this system size, there are better optima that have not been found in this number of descents. The data show that for all the

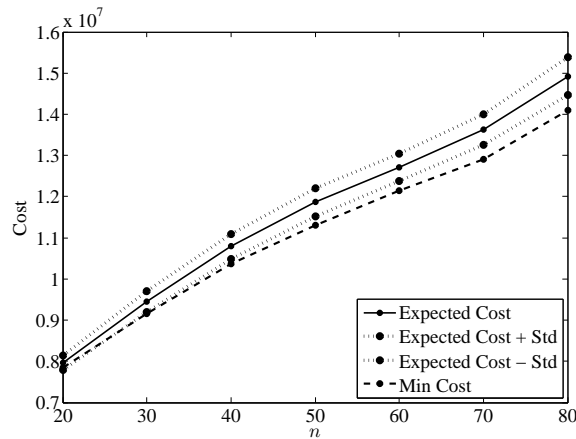


Figure 21: Plot of the expected minimum cost and the average cost of the local optima found by 3-opt versus  $n$ . We have computed the standard deviation in the cost and show the expected cost plus and minus 1 standard deviation. This is found by performing  $10^5$  hill-climbs on 20 randomly generated instances of the problem for Sup-Drilling problem.

problem types as the system size grows the probability of finding the global optimum solution decreases exponentially, although with different rates.

Figure 21 shows that the gap between the expected cost and minimum cost increases with the system size and figure 22 shows the probability of finding the global optimum decreases exponentially with the system size. These two figures show two different aspects of the same property of the fitness landscape. This property shows how the problem becomes harder as the system size grows. In order to show this for all the problem types, figure 23 shows the gradient of the fitting line in figure 22 versus the increasing rate of the gap in figure 21 with the system size. The data clearly show the correlation between the gap and the probability of finding the global optimum. The Job-Scheduling problem and the Random problem can be considered as the hardest problems, as the decay rate of the probability of finding global optimum is the highest among all the problem types. When the Random problems become symmetric or closed, the decay rate decreases, meaning the Random Symmetric and the Random Closed problems are easier than completely Random problems. Among these problem types, the Additive-Drilling and the Euclidean problem which show similar behaviour are the two easiest problems.

### 3.6 Distance Between Optima

One important property of the fitness landscape is the distance between the local optima which shows how the local optima are correlated. The distance between two solutions is simply defined as the number of non common edges in the two solutions (note we are using here the Hamming distance rather than the number of 3-opt moves between configurations). For symmetric problems the edges are undirected, so both the directions of the edges are considered the same. The histogram of the distances between local optima for two different problem types is shown in figure 24. The results are averaged over 20 randomly drawn problem instances for  $n=50$ . The interesting property of the histogram are the double peaks in the histogram of Additive-Drilling problem, one higher and wider peak at the left, and one lower and thinner at the right side. This peculiar property is because of the nature of this problem. As described in section 2.1.6, in the definition of the Drilling problem, the energy needed for the machine to move the drill down is

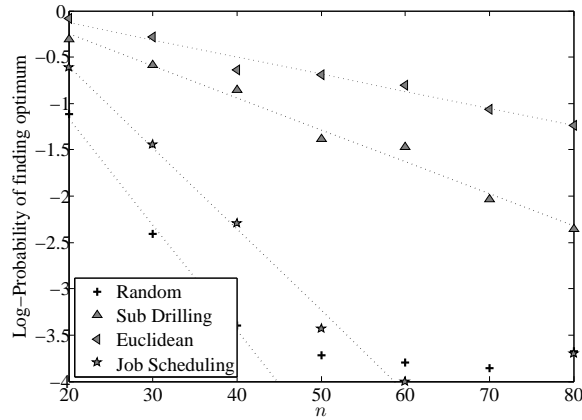


Figure 22: Log-Probability of finding the lowest cost local optimum for four different problem types versus system size  $n$ . The number of hill-climbs is  $10^4$  and the data are averaged over 20 different random problem instances.

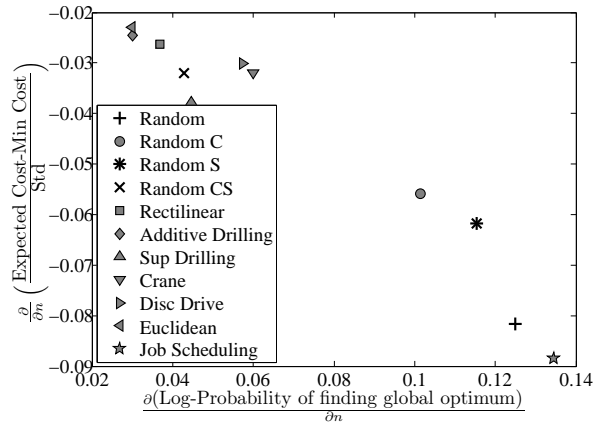


Figure 23: The increasing rate of the gap between the expected cost and cost of the global optimum with  $n$  versus the decay rate of the probability of getting to the global optimum with  $n$ . The data are found for  $n=20$  to  $n=80$  with the step of 10 for 20 different problems and  $10^4$  descents on each.



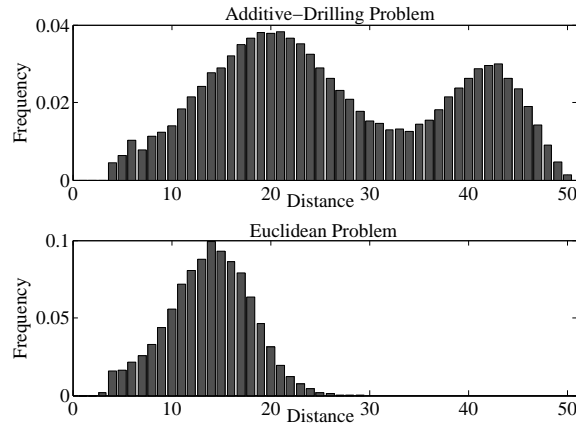


Figure 24: Histogram of the distances between the local optima for two different problem types. The size of the problem is  $n=50$ , and the results are averaged over 20 different randomly drawn problem instances. The number of hill-climbs is  $10^5$  on each problem instance.

zero. Due to this property, although this problem is not a symmetric problem, some sub-tours in a solution can be reversed without posing a significant change to the cost of the solution. The change to the cost of a solution posed by reversing a sub-tour is the difference in the  $y$ -coordinate of the starting and the ending point of the sub-tour. In this sense the problem can be considered as a partially symmetric problem. This property causes the peculiar behaviour in the histogram of the distances between the local optima.

The histogram of the distances has a bell shape in the Euclidean problem. This bell shape behaviour of the histogram is seen in all the problem types except the Additive-Drilling and in some degrees in the Sup-Drilling problem. In all the problem types, the peak of the histogram is located at the left side of the expected distance between two randomly drawn solutions. Note that for the size of  $n=50$ , the expected distance between two randomly drawn solutions is around 49.5 (we have found this by randomly generating  $10^9$  pair of configurations and calculating the average distance). Therefore the histograms of the figure 24 clearly suggest that the local optima are much closer to each other than the expected distance between two random solutions.

Figure 24 shows the histogram of distances for all local optima irrespective of their cost. It is also interesting to see how the distance between local optima depends on their cost. We show this information for the Sup-Drilling problem in figure 25 where a 2-dimensional histogram is plotted. We observe that the best (lowest cost) local optima are closer to each other than the less fit local optima. Half of the best 10% of local optima are within a distance of 5 from each other. We observe that the less fit local optima are much further from each other than the best fit optima.

### 3.7 Distance to Global Optimum

The measure of the mean distance to the global optimum from a local optimum of cost  $c$  for different size of the problem for Euclidean problem is shown in figure 26. The scaling in the vertical axis demonstrates that the distances from a local optimum to a global optimum scales linearly with the problem size. The scaling behaviour of the curve is the same for all the problem types. Making the last step from an optimum of cost  $c_{min} + \epsilon$  to  $c_{min}$  clearly grows with  $n$ , although it is difficult to be sure how this distance grows. This question depends on extrapolating

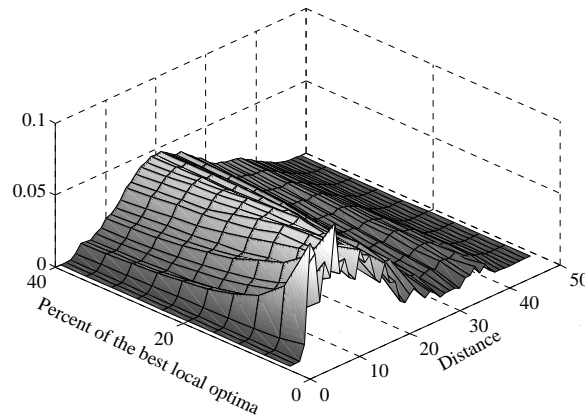


Figure 25: Histogram of the distances between the best local optima for different number of local optima for Sup-Drilling problem. The size of the problem is  $n=50$ . The data are averaged over 20 different problem instances and  $10^5$  descents on each.

the curve in figure 26 to the y-axis.

Another way of viewing this information is as a density plot of the cost versus distance to the global optimum. We show these plots for a Euclidean problem and a Random problem in figure 27. There is clearly a strong fitness distance correlation for the Euclidean problem, but very little for the Random problem. Many population based algorithms attempt to exploit the fitness distance correlation. Clearly, there is some hope in case of the Euclidean problems, but much less hope for Random TSP problems.

### 3.8 Expected Cost in Hamming Sphere

We noted earlier that all the problems had a relatively slow decline in the autocorrelation function, corresponding to a large correlation length. This is a result of the cost being the sum of all the edges in the tour. If we make a local move which changes only a few edges then there will be only a relatively small change in the cost. To examine this we study the mean cost in a Hamming sphere from a configuration. The Hamming sphere of radius  $h$  being the set of configurations that differ from the start configuration by exactly  $h$  edges.

Figure 28 shows the expected cost of a configuration in a distance sphere of radius  $h$  from a global optimum. To construct the graph, we performed some random number of 3-opt moves to the global optimum and then measured the distance from the global optimum. The process was repeated  $10^5$  times and the costs at each Hamming distance averaged.

Interestingly, we observe that the cost increases linearly with the distance up to a Hamming distance of  $n - 1$ . We note that the average Hamming distance between random tours is around  $n - 1$ . We also show the expected cost of a random tour in figure 28 as a horizontal dashed line. A very simple model for the expected cost in a Hamming sphere of radius  $h$  about a tour  $s$  is

$$c(s, h) \approx \left(1 - \frac{h}{n}\right) c(s) + \frac{h}{n} \bar{c}$$

where  $c(s)$  is the cost of the current tour and  $\bar{c}$  is the average cost.

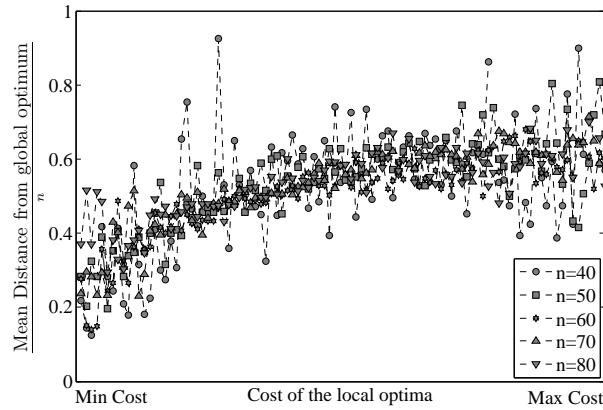


Figure 26: Measure of the mean distance to the global optimum from a local optimum of cost  $c$  for different size of the problem. This is for Euclidean problem, averaged over 20 problem instances.

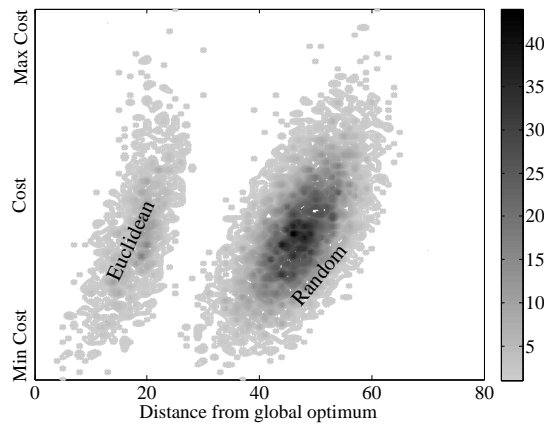


Figure 27: Density plot of local optima as a function of their distances from the most frequently visited global optimum. The shading of the point shows the number of local optima at that cost and distance. The size of the problem is  $n=80$  and this is for two problem types, Euclidean (left side) and Random (right side) problems. The number of Hill-Climbs is  $10^5$ .

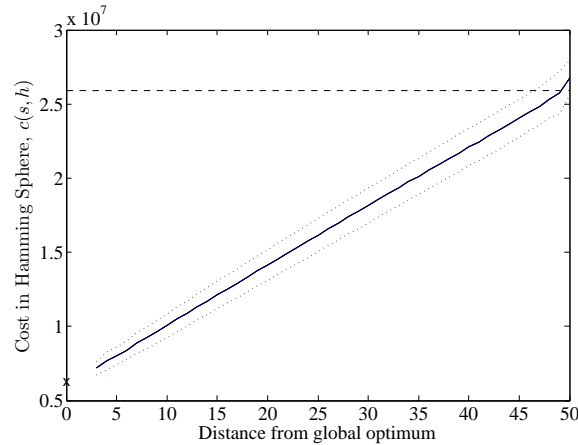


Figure 28: Expected cost of configurations in distance sphere of radius  $h$  around a global optimum for  $n=50$ . This is for Euclidean problem. The dotted lines show one standard deviation around the mean. The dashed line shows the expected cost of the random solutions.

### 3.9 Return Probabilities

Another interesting property is the probability of returning to a local optimum starting at a fixed Hamming distance from the optimum. We can empirically measure the basin of attraction of a local optimum by repeatedly jumping to a configuration at a certain distance from the local optima and running the local search algorithm until it reaches a local optimum. We then see if this is the same local optimum from which we started. We can thus measure the return probability starting from a configuration in a given distance sphere. In figure 29 we show these return probabilities for four problems starting from the global optimum using 3-opt moves. The curves are not monotonic because 3-opt moves can change some number edges easier than others. For example, it is easy to change 3 edges than 5 edges. As we would expect the return probabilities falls off faster as the problems become harder.

## 4 Principal Component Analysis

Although we have seen that the number of local optima grows exponentially with the system size, many of these may be formed by combining other optimal tours. For example, we could have a tour consisting of 10 segments. There may be two routes through the cities in each segment both of which are local optima. In this case there would be  $2^{10} = 1024$  local optima which are a consequence of these 10 binary choices. To investigate the underlying variety in the local optima we have used principal component analysis (PCA) to find the subspaces with a large variation. To do this we consider each tour as a binary vector in the space of edges. That is, for a symmetric problem the tour is represented as a binary vector consisting of  $n$  non-zero components in a total of  $n(n-1)/2$  possible components. Similarly for asymmetric problems the tour is represented as a binary vector consisting of  $n$  non-zero components in a total of  $n(n-1)$  possible components. To perform PCA we subtract the mean vector of all the optima and then compute the eigenvalues and eigenvectors of the covariance matrix (we can compute this efficiently using singular value decomposition).

To illustrate this approach we show the average solution of the local optima in figure 30 for a Euclidean problem with 30 cities. The width of the edges shows the number of times they appear in different local optima. It is clear from the graph that there are many edges that do

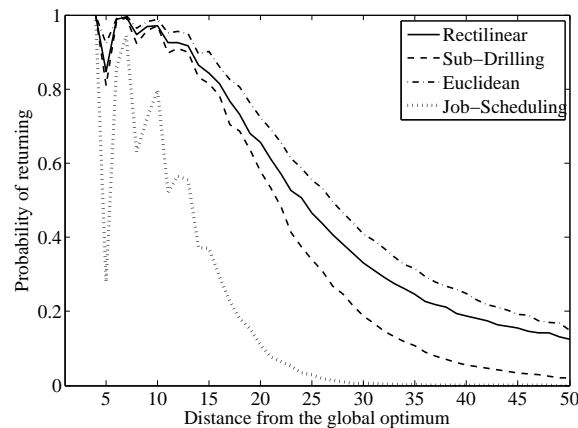


Figure 29: The return probability starting from different distances from the global optima. This is for four different problem types for size of the problem  $n=50$ .

not appear in any of the local optima, and there are some which appear in all the local optima. This property means that the search algorithms do not need to involve such edges in their search process, and a huge proportion of the landscape can be ignored in the search process.

The eigenvectors correspond to orthogonal directions in edge space. The eigenvectors with larger eigenvalues correspond to directions where there is a large variance. That is, the local optima vary significantly in these directions. We can think of these eigenvectors or “eigentours” as showing the most significant way in which the local optima differ. In figure 31 we show the eigenvector corresponding to the largest eigenvalue (i.e. the principle component) for the same problem shown in figure 30. Some components are positive and others are negative. We can (roughly) interpret an eigentour as a choice between either choosing the positive edges and removing the negative edges or the other way around.

By plotting the eigenvalues against their rank we can see the dimensionality of the subspace where there is a significant variation between local optima. The size of the eigenvalue is equal to the mean squared reconstruction error of the optima if we project out that dimension. In figure 32, we have plotted the spectrum of eigenvalues for four instances from different problem types. We see that the degree of variation in the Euclidean and the Rectilinear problems we have a relatively small number of non-zero eigenvalues and the variance is small (indicating that the eigentours do not involve that many edges). In contrast, the Scheduling and Sup-Drilling problems show many more directions of variation and considerably larger variations.

One final use of PCA is that it allows us to project the local optimum solution into a meaningful low dimensional subspace, allowing some visualisation of the structure of the local optima. We have done this for six problems in figure 33 where we have plotted the local optima into the space spanned by the first two principle eigenvectors (the eigenvectors with the two largest eigenvalues). Furthermore, we have plotted the data so that the size of the dots represents the fitness of the local optima; the better local optima are represented with larger dots.

The first graph is for a Random problem instance, where there is no particular structure in the local optima. For all the problem instances of the Crane problem we almost always see a squared clustered structure for local optima. The squares may form any angle with respect to the horizontal line or may consist of different number of clusters for different problem instances, but the common feature among all the problem types (except Random problems) and problem

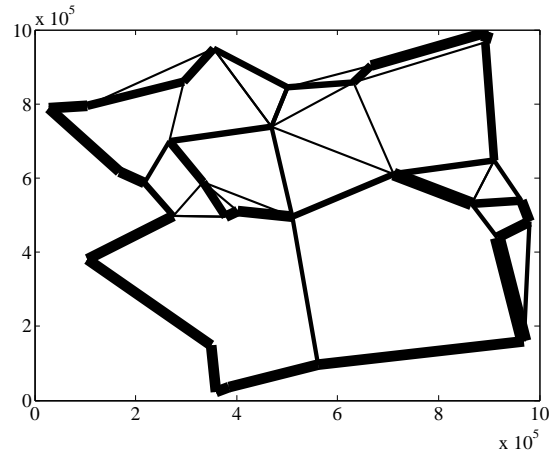


Figure 30: The schematic shape of the average of local optima solutions for a Euclidean problem with 30 cities. The width of the edges shows the number of times they appear in different local optima.

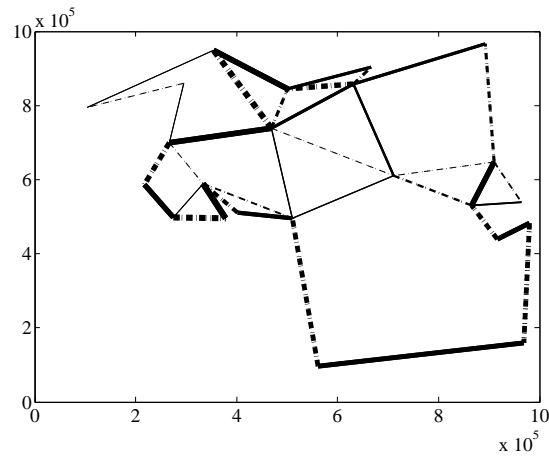


Figure 31: The schematic shape of the eigenvector of the local optima solutions with highest eigenvalue for a Euclidean problem with 30 cities. The width of the edges shows the values in the eigenvector. The lines represent the positive and the dotted lines represent the negative values.

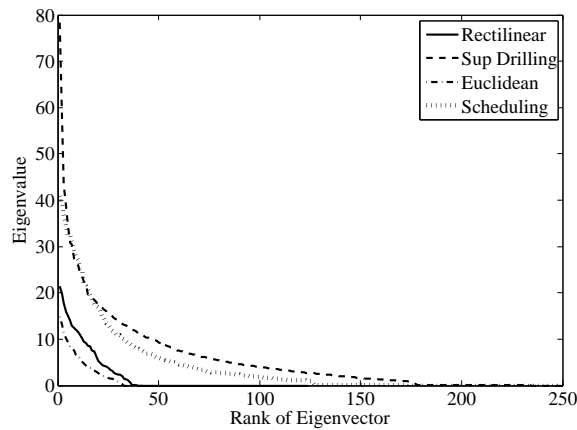


Figure 32: The eigenvalues of the eigenvectors of the edges in local optima in a descending order. This is for four different problem types for size of the problem  $n=50$ .

instances is the fitness of the local optima form clusters. We see some of the clusters consisting of better local optima and some other containing inferior local optima solutions. The third graph is for Euclidean problem, of which the projected local optima usually form two parallel elongated football shape clusters. But not all the problem instances show exactly the same behaviour, the angle of the clusters to the horizontal line differs from problem instance to another, the clusters are occasionally not parallel to each other and sometimes there are more than two clusters. For Additive-Drilling problem we see a similar behaviour to the Euclidean problem, but almost always the number of the clusters is two (in over 50 problem instances we studied) and the gap between the clusters is much wider. The next graph shows the Disk-Drive problem, where the local optima usually form parallel lines of clusters. In this problem type the local optima sometimes form two or more parallel lines and in some cases 10 parallel lines. The lines are sometimes close and mixed together such that they form a square shape. And finally the last graph shows the Job-Scheduling problem, where for almost all the problem instances we observed (for more than 50 problem instances with different sizes), the local optima form squares, although the squares sometimes consist of some clusters.

## 5 Discussion

In this paper we use 3-opt for our studies. Clearly, many of the properties we report in this paper depend on the neighbourhood structure and the local search operator. This is particularly true for properties like auto-correlation that measure the local ruggedness of the landscape via random walk. However, the degrees to which these properties depend on the local search operator vary from a property to another. For example, using an operator that takes longer steps and offers larger number of neighbours to a solution (like 4-opt) would change some of the data presented in this paper. For time to local optima and the plateaux for example, choosing such an operator would cause an operator to take less number of steps to reach a local optimum. Also the operator may escape plateaux more easily, so the size of the plateaux might decrease. Using such an operator also decreases the number of local optima in the landscape as many of the basins of attractions merge together to form a single basin of attraction. For the same reason, probability of finding a local optimum or the global optimum would increase.

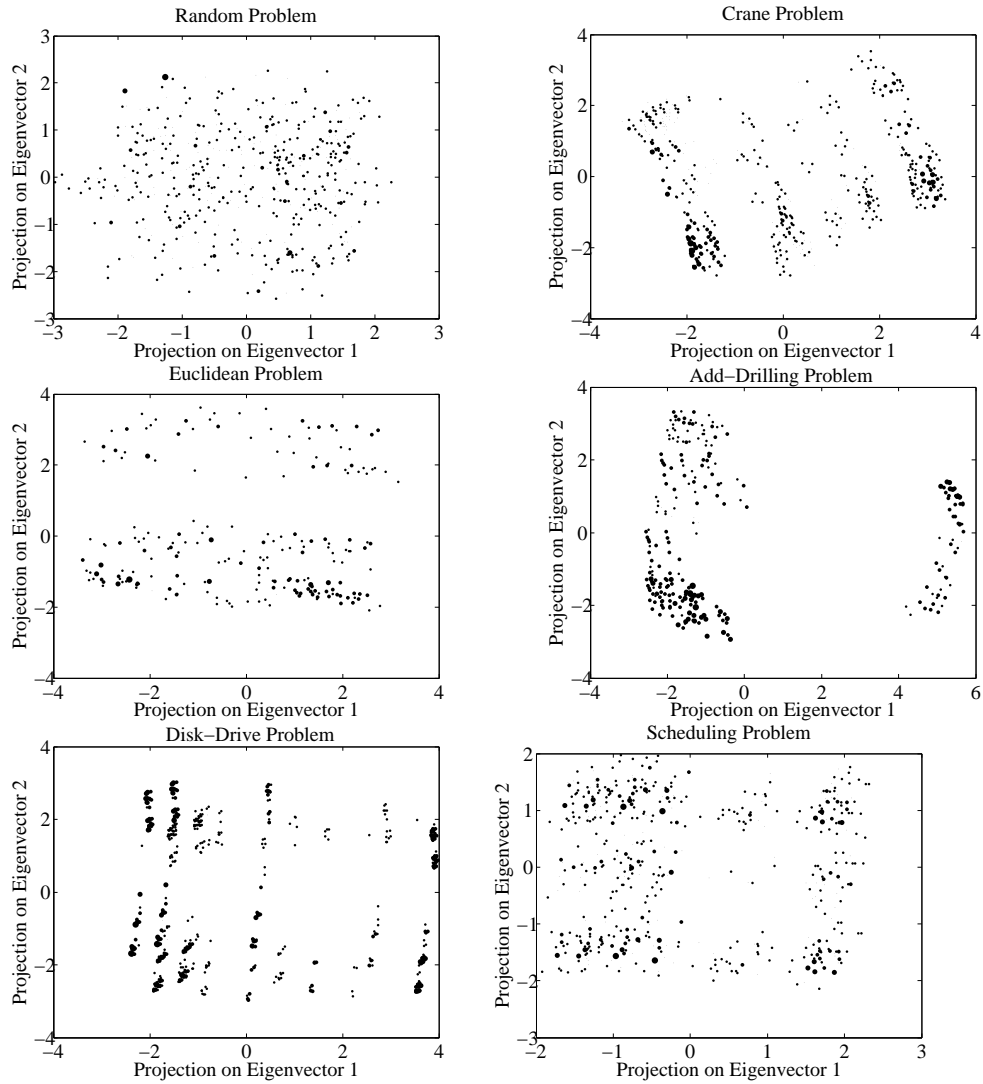


Figure 33: The projection of the local optima solutions into two dimensions using the two eigenvectors with the highest eigenvalues for three problem types.



Although this does not imply that such an operator is better to use, as each step of the algorithm is more time consuming.

We believe some properties, however, would not change if different operators were used. These properties include the shape of the distribution of local optima at each cost level, the property that the probability of finding the global optimum decreases exponentially with the problem size, the property that the closer optima to the global optima have, in expectation, lower cost and the PCA properties. We believe these properties reflect the nature of these problems and so are operator independent. We have two major reasons to believe this. First these properties are also seen on other different NP-Hard combinatorial problems, including Max-Sat Prügel-Bennett and Tayarani-N. (2011), Graph-Colouring Tayarani and Prügel Bennett (2014) and Quadratic Assignment Tayarani and Prugel-Bennett (2013) Problems.

The second reason is that some properties like the relationship between the probability of finding the global optimum and time to local optima should not change if different operator were used. To explain it more clearly, let assume the following example. In this paper we showed that time to local optima increases linearly and the probability of finding the global optimum increases exponentially with the problem size. So solving the problem becomes exponentially more time consuming as the problem size increases. This should be the case for any operator. Let assume that there is an operator that if used, the probability of finding the global optimum increase polynomially and time to local optima increase linearly with problem size. Then it would mean that the time needed to keep the expected number of times the global optimum is found equal to one, grows polynomially with problem size. So in one sense, NP-Hard problems are solved polynomially. Although this might be the case, we believe that there are more evidence for believing the opposite.

The PCA properties, and the property that the closer local optima to the global optimum are, on average, fitter than further ones are other examples of properties that we believe are local search operator independent. Because these show the structure of the problems and the relationship between good solutions and it does not matter how these good solutions are found.

The other issue that can affect the analyses in this paper is the isotropy of the landscape. In Greenwood and Hu (1998) it is shown that studying anisotropic landscapes requires some considerations. A landscape is anisotropic if the fitness of solutions in some regions in the landscape are considerably different from other regions. This, for example, happens when some regions in the landscape do not satisfy some problem constraints and are remarkably at lower fitness compared to other regions. These low cost regions are called sink hole regions. For this type of landscapes, an analysis that uses a long random walk to estimate ruggedness or correlation does not provide accurate results, as some walks may hit sink holes. For several reasons we believe that many of our analysis, in terms of isotropy of the landscape, are valid. First, as discussed earlier in this paper, we believe such sink holes do not exist in the landscape of TSP. Second, to study the proposed properties in this paper, we do not use one single long random walk. We use a large number of local search starting from different configurations (although properties like auto-correlation are prone to this problem, and this is why we believe these type of properties are not true representative of problem hardness). Third, even if there were sink hole regions in the fitness landscape, they would not affect many of the results presented in this paper. Sink holes affect random walks because when they pass over the holes, observe different ruggedness. Our local search on the other hand always chooses the best neighbours; thus it never passes over a sink unless it starts from one.

Even if the search starts from a sink, it still has good chance of escaping from it and finding a local optimum outside the sink.

## 6 Conclusion

Fitness landscapes are complicated objects which are not easily characterised by a small number of properties. In consequence we have to settle for a drawn out conclusion. For the sake of clarity we enumerate some of the most striking properties below.

1. For TSP the number of steps to reach a minimum appears to grow linearly for most problem types. However, each step of 3-opt naively takes  $O(n^3)$  time to compute (as we potentially have to search all edge triples to find one that can be optimised), this can be time consuming. This does not, in itself, make the problem difficult. What makes the problems difficult is that the local optimum we find is unlikely to be the global optimum.
2. For all problem types we observed that the number of local optima grows exponentially as some function of the size of the instance, but this rate of this growth varied considerably between problem types.
3. In most TSP types that we examined the fitter local optima tended to have a significantly larger probability of being found than the less fit local optima. Thus, the number of local optima does not necessarily determine the probability of finding the global optimum.
4. The probability of reaching a global optimum decreases exponentially with the system size. This property is correlated with a widening gap between the cost of the global optimum and the expected cost of an optimum found by local search, measured in units of the number of standard deviations in the cost of the optima found by local search.
5. There is a correlation between the fitness of a local optimum and its distance to the global optimum with fitter solutions closer on average to the less fit solutions. This behaviour appears to scale linearly with the problem size. However, this correlation is very dependent on the problem type and the correlation is especially low for Random problems.
6. The difference between the cost of a reference configuration and the average cost of the set of configuration at a fixed Hamming distance,  $h$ , from the reference is almost exactly linearly correlated with  $h$ . This is a result of the fact that the objective function consists of a sum of terms, where each term consists of only a few variables. Thus when we change a small number of variables we only produce a small change in the cost. This function is almost exactly linear between the cost of the tour we start from and the expected cost of random tour at a Hamming distance of  $n$ . This long-range correlation means that there is a considerable similarity in the long-range structure of the fitness landscape.

Although, it is not the intention of this paper to make a comparison with other problems, it is perhaps worth noting in passing that very similar properties to those described above were also observed in MAXSAT Prügel-Bennett and Tayarani-N. (2011) and graph-colouring Tayarani and Prügel Bennett (2014). The current paper adds new texture to our description of fitness landscapes by showing how the properties vary between different problem types. Most of the properties of the landscape that we have analysed are qualitatively similar for all problem types, but they vary significantly quantitatively. In particular, the more random problems tend to have significantly more local optima, with the cost of the local optima being significantly less correlated with its proximity to the global optimum tour. One important lesson of this analysis is that Euclidean TSP is exceptional in the set of problem types we analysed in so far as it is relatively

easy to solve. One should therefore be on ones guard in extrapolating results from Euclidean TSP to more general TSP problems.

A final contribution of this paper was the introduction of PCA to analysing the structure of the fitness landscape. This technique provides a qualitative understanding of the degrees of variability between local optima in terms of the spectrum of eigenvalues. It clearly shows that although there is an exponential number of local optima, they are actually many fewer original ways in which the local optima differ from each other. Again this is a property which depends on the problem type. We have also seen that PCA can provide a means of visualising the structure of the local optima by allowing a meaningful projection to a low-dimensional subspace. This often reveals considerable clustering in the local optima which is not transparent otherwise. Although we have shown PCA results for small problem instances, it is not limited to small instances. The complexity of PCA depends linearly on the number of edges that differ between the local optima and cubically on the number of local optima. Thus, provided the number of local optima found is not too large, it is feasible to perform PCA on large problems. We believe that PCA could be used to direct the search of a new algorithm. The usefulness of PCA is clearly problem dependent, but as we saw in the Crane example in figure 33, the fit optima are often confined to particular clusters. PCA does not require the solutions to be optimal solutions and can just use good solutions returned by an optimiser. Although it may seem expensive to compute many good solutions, on large problems this may be necessary anyway. Finally, it is worth pointing out that PCA is not limited to TSP but can be used on pretty much any problem.

## References

- Alander, J., Zinchenko, L., and Sorokin, S. (2002). Analysis of fitness landscape properties for evolutionary antenna design. In *Artificial Intelligence Systems, 2002. (ICAIS 2002). 2002 IEEE International Conference on*, pages 363 – 368.
- Alander, J. T. (1999). *Practical Handbook of Genetic Algorithms: Complex Coding Systems, Chapter 13, Population size, building blocks, fitness landscape and genetic algorithm search efficiency in combinatorial optimization: An empirical study*, volume 3. CRC press.
- Altenberg, L. (1997). Fitness distance correlation analysis: An instructive counterexample. In Bäck, T., editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 57–64, San Mateo, CA. Morgan Kaufmann.
- Angel, E. and Zissimopoulos, V. (1998). Autocorrelation coefficient for the graph bipartitioning problem. *Theoretical Computer Science*, 191:229–243.
- Arora, S. (1998). Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *J. ACM*, 45(5):753–782.
- Boese, K., Kahng, A., and Muddu, S. (1994). On the big valley and adaptive multi-start for discrete global optimizations. *Operation Research Letters*, 16(2).
- Boese, K. D. (1995). Cost versus distance in the travelling salesman problem. Technical report, UCLA computer science department, Los Angeles.
- Bouziri, H., Mellouli, K., and Talbi, E.-G. (2011). The  $k$ -coloring fitness landscape. *Journal of Combinatorial Optimization*, 21:306–329.
- Cirasella, J., Johnson, D. S., McGeoch, L. A., and Zhang, W. (2001). The asymmetric traveling salesman problem: Algorithms, instance generators, and tests. In *Revised Papers from the Third International Workshop on Algorithm Engineering and Experimentation, ALENEX '01*, pages 32–59. Springer-Verlag.

- Collard, P., Vérel, S., and Clergue, M. (2007). Local search heuristics: Fitness cloud versus fitness landscape. *CoRR*, abs/0709.4010.
- Czogalla, J. and Fink, A. (2011). Fitness landscape analysis for the no-wait flow-shop scheduling problem. *Journal of Heuristics*, pages 1–27.
- DR Hains, L. W. and Howe, A. (2011). Revisiting the big valley search space structure in the tsp. *Journal of the Operational Research Society*, 62:305 – 312.
- Fonlupt, C., Robilliard, D., and Preux, P. (1997). Fitness landscape and the behavior of heuristics. In *Evolution Artificielle 97 (EA'97)*.
- Forrest, S. and Mitchell, M. (1993). What makes a problem hard for a genetic algorithm? some anomalous results and their explanation. Technical Report 3.
- Greenwood, G. W. and Hu, X. S. (1998). On the use of random walks to estimate correlation in fitness landscapes. *Computational Statistics & Data Analysis*, 28(2):131 – 137.
- Grover, L. K. (1992). Local search and the local structure of NP-complete problems. *Operations Research Letters*, 12:235–243.
- Hertz, A., Jaumard, B., and de Aragao, M. P. (1994). Local optima topology for  $k$ -coloring problem. *Discrete Applied Mathematics*, 49:257–280.
- Horn, J. and Goldberg, D. E. (1995). Genetic algorithm difficulty and the modality of fitness landscapes. *Foundations of Genetic Algorithms*, 3.
- Hornby, G. S. (1996). The recombination operator, its correlation to the fitness landscape and search performance. In *Master of Science Thesis*. University of Alberta.
- Hoshino, T., Mitsumoto, D., and Nagano, T. (1998). Fractal fitness landscape and loss of robustness in evolutionary robot navigation. *Auton. Robots*, 5:199–213.
- Huang, D., Shen, Z., Miao, C., and Leung, C. (2009). Fitness landscape analysis for resource allocation in multiuser ofdm based cognitive radio systems. *SIGMOBILE Mob. Comput. Commun. Rev.*, 13:26–36.
- Jones, T. and Forrest, S. (1995). Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 184–192, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Lehn, R. and Kuntz, P. (2001). A contribution to the study of the fitness landscape for a graph drawing problem. In Boers, E., editor, *Applications of Evolutionary Computing*, volume 2037 of *Lecture Notes in Computer Science*, pages 172–181. Springer Berlin / Heidelberg.
- Lu, G., Li, J., and Yao, X. (2011). Fitness-probability cloud and a measure of problem hardness for evolutionary algorithms. In *Proceedings of the 11th European conference on Evolutionary computation in combinatorial optimization*, EvoCOP' 11, pages 108–117, Berlin, Heidelberg. Springer-Verlag.
- Martin, W., Barker, A., and Cohoon, J. (1999). Problem perturbation: implications on the fitness landscape. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*.

- Mathias, K. and Whitley, D. (1992). Genetic operators, the fitness landscape and the traveling salesman problem. In *Parallel Problem Solving from Nature*, pages 219–228. Elsevier Science Publishers.
- Merz, P. (2004). Advanced fitness landscape analysis and the performance of memetic algorithms. *Evol. Comput.*, 12:303–325.
- Merz, P. and B.Freisleben (2000). Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. *IEEE Transactions on Evolutionary Computation*, 4(4):337 – 352.
- Merz, P. and Freisleben, B. (1998). Memetic algorithms and the fitness landscape of the graph bipartitioning problem. In *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*, PPSN V, pages 765–774. Springer-Verlag.
- Pošík, P. and Franc, V. (2007). Estimation of fitness landscape contours in eas. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, GECCO '07, pages 562–569, New York, NY, USA. ACM.
- Prügel-Bennett, A. and Tayarani-N., M.-H. (2011). Maximum satisfiability: Anatomy of the fitness landscape for a hard combinatorial optimisation problem. *IEEE Transactions on Evolutionary Computation*, 15:–.
- Qasem, M. and Prügel-Bennett, A. (2010). Learning the large-scale structure of the max-sat landscape using populations. *IEEE Transactions on Evolutionary Computation*, 14(4):518–529.
- Rardin, R. L., Tovey, C., Pilcher, M., and Pardalos, P. (1993). Analysis of a random cut test instance generator for the tsp. In *World Scientific*.
- Ratle, A. (1998). Accelerating the convergence of evolutionary algorithms by fitness landscape approximation. In *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*, PPSN V, pages 87–96, London, UK. Springer-Verlag.
- Riley, J. and Ciesielski, V. (2010). Fitness landscape analysis for evolutionary non-photorealistic rendering. In *Proceedings of IEEE World Congress on Computational Intelligence*, Barcelona.
- Shen, L. and He, J. (2010). A mixed strategy for evolutionary programming based on local fitness landscape. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8.
- Stadler, P. (1995). Towards a theory of landscapes. *Complex Systems and Binary Networks*, pages 78–163.
- Stadler, P. (1996). Landscapes and their correlation functions. *Journal of Mathematical Chemistry*, 20(1):1–45.
- Stadler, P. F. and Schnabl, W. (1992). The landscape of the traveling salesman problem. *Physics Letters A*, 161(4):337 – 344.
- Suzuki, H. and Iwasa, Y. (1997). Ga performance in a babel-like fitness landscape. In *Tools with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on*, pages 357–366.

- Tavares, J., Pereira, F. B., and Costa, E. (2006). The role of representation on the multidimensional knapsack problem by means of fitness landscape analysis. In *Proceedings of IEEE Congress on Evolutionary Computation Sheraton, Vancouver*.
- Tayarani, M. and Prugel-Bennett, A. (2013). On the landscape of combinatorial optimisation problems. *Evolutionary Computation, IEEE Transactions on*, PP(99):1–1.
- Tayarani, M. H. and Prügel Bennett, A. (2014). Anatomy of the fitness landscape for graph-colouring problem. *Accepted for publication in Journal of Swarm and Evolutionary Computation, Elsevier*, 10:1.
- Vanneschi, L., Tomassini, M., Collard, P., Vrel, S., Pirola, Y., and Mauri, G. (2007). A comprehensive view of fitness landscapes with neutrality and fitness clouds. In *Genetic Programming*, volume 4445 of *Lecture Notes in Computer Science*, pages 241–250. Springer Berlin - Heidelberg.
- Vrel, S., Collard, P., Tomassini, M., and Vanneschi, L. (2007). Fitness landscape of the cellular automata majority problem: View from the "olympus". *Theor. Comput. Sci.*, 378:54–77.
- Vérel, S., Collard, P., Tomassini, M., and Vanneschi, L. (2008). Neutral fitness landscape in the cellular automata majority problem. *CoRR*, abs/0803.4240.
- Watson, J.-P. (2010). An introduction to fitness landscape analysis and cost models for local search. *HANDBOOK OF METAHEURISTICS*, 146:599–623.
- Weinberger, E. D. (1990). Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biol. Cyber.*, 63:325–336.
- Wiles, J. and Tonkes, B. (2006). Hyperspace geography: Visualizing fitness landscapes beyond 4d. *Artif. Life*, 12:211–216.
- Wright, S. (1932). The roles of mutation, inbreeding, crossbreeding, and selection in evolution. In *Proceedings of 6th Congress of Genetics*, volume 1, page 365. ACM Press.
- Wu, Y., McCall, J., and Corne, D. (2011). Fitness landscape analysis of bayesian network structure learning. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 981–988.
- Zhang, W. (2004). Configuration landscape analysis and backbone guided local search: part i: Satisfiability and maximum satisfiability. *Artif. Intell.*, 158:1–26.
- Zhang, W., Rangan, A., and Looks, M. (2003). Backbone guided local search for maximum satisfiability. In *Proc. of the 18th Intern. Joint Conference on Artificial Intelligence*, pages 1179–84.