

# On the Landscape of Combinatorial Optimisation Problems

Mohammad-H. Tayarani-N. and Adam Prügel-Bennett

**Abstract**—This paper carries out a comparison of the fitness landscape for four classic optimisation problems: Max-Sat, Graph-Colouring, Travelling Salesman and Quadratic Assignment. We have focused on two types of properties, local average properties of the landscape and properties of the local optima. For the local optima we give a fairly comprehensive description of the properties, including the expected time to reach a local optimum, the number of local optima at different cost levels, the distance between optima, and the expected probability of reaching the optima. Principle component analysis is used to understand the correlations between the local optima. Most of the properties we examine have not been studied previously, particularly those concerned with properties of the local optima. We compare and contrast the behaviour of the four different problems. Although, the problems are very different at the low level, many of the long-range properties exhibit a remarkable degree of similarity.

**Index Terms**—Combinatorial Optimisation Problems, Graph-Colouring problem, Travelling Salesman problem, Maximum Satisfiability problem, Quadratic Assignment problem, fitness landscape, scaling analysis, long-range correlation.

## I. INTRODUCTION

TO design and improve heuristic search algorithms for combinatorial optimisation problems requires an understanding of the fitness landscapes for these problems. There has thus been a considerable interest in study of fitness landscapes. However, most of these studies have focused on a relatively small number of properties such as the autocorrelation [1] and fitness distance correlation [2], often in the hope that such measures may capture the problem difficulty. Alas, we have seen little evidence that a very small number of measures can capture the complexity of combinatorial optimisation problems. It is our belief that to obtain a thorough understanding of these problems requires examining a much broader set of properties of the fitness landscapes. Thus in a series of papers we have undertaken a detailed study of four well known combinatorial optimisation problems: maximum satisfiability (Max-Sat) [3], graph-colouring [4], the Travelling Salesman Problem (TSP) [5] and the quadratic assignment problem [6]. This paper gives a comparison of these four problems.

On the surface, these problems look very different. For example the natural search space for Max-Sat is binary vectors, while that of graph-colouring is the set of  $k$ -ary vectors, where  $k$  is the number of colours, and TSP and quadratic assignment

are most naturally, encoded as permutations. The problems have different symmetries, so that Max-Sat and quadratic assignment have no obvious symmetry, symmetric TSP problems have a simple symmetry under reversing the direction of the tour (although this is only a two fold symmetry, it has a profound consequence for search operators) and finally graph-colouring has a  $k!$ -fold colour permutation symmetry, where  $k$  is the number of colours. The search operators also vary considerably between problems, particularly for TSP where finding a move operator is non-trivial. As the search operators determine the neighbourhood structure of the landscape, each of the problems we have studied has a different neighbourhood topology.

At a more profound level, Max-Sat and graph-colouring are constraint satisfaction problems (that is, their objective function can be seen as a sum of constraints). In both problems we can progressively add constraints. With a small number of constraints, it is possible to find configurations which satisfy all the constraints (giving zero cost solutions). As more constraints are added, eventually a situation is reached when it is no longer possible to satisfy all the constraints, and the best an optimisation algorithm can do is to attempt to find a solution that satisfies as many constraints as possible. If the instances are drawn uniformly from some ensemble of problems, it is typical to find that the probability of choosing a problem that satisfies all the constraints makes a very rapid jump, which has long been identified as a phase transition from satisfiability to unsatisfiability. Empirically, it is found that the time complexity of the problem changes around this phase-transition. Thus, below and away from the phase transition, the problems are found to be easy to solve. Around the phase transition the problems are difficult to solve. Away and above the phase transition, it is often easy to prove that the problem is unsatisfiable, but it tends to be difficult to find the lowest cost solutions. We will see that this phase transition determines other properties of the fitness landscape, in particular it is associated with large plateau regions which can significantly slow down local search. TSP and quadratic assignment are not constraint satisfaction problems and consequently do not undergo a satisfiability phase transition.

Despite the many differences between these combinatorial optimisation problems, we will see that there is a considerable degree of similarity in the large scale structure of the fitness landscape. These similarities provide an explanation why general purpose algorithms such as evolutionary algorithms have quite wide applicability. One explanation of the similarity between problems is that their cost functions all involve a sum of terms, each involving a small number of variables. For a

Mohammad-H. Tayarani-N. is with the Department of Electrical and Computer Science, University of Southampton, Southampton, UK email: mhtn1g09@ecs.soton.ac.uk

Adam Prügel-Bennett is with the Department of Electrical and Computer Science, University of Southampton, Southampton, UK email: apb@ecs.soton.ac.uk

randomly chosen set of variables the cost can be seen as a sum of rather weakly correlated random variables; however, neighbouring configurations are relatively highly correlated, leading to quite smooth landscapes with long-range correlations. These properties appear to lead to the similarity in the large scale behaviour of the fitness landscape that we observe.

Clearly, there is considerable selectivity in the problems we have chosen to study. For example, it is possible to envisage problems where the objective function is again a sum of a large number of terms, but the terms are much more strongly coupled. Examples of this are the Sherrington-Kirkpatrick spin-glass model [7] and the binary perceptron [8], [9], [10]. In such models we would expect the fitness landscape to be locally much less smooth, although there would still exist long-range correlations. We have chosen not to study such models as we believe the ones that we have chosen are more representative of the type of classical optimisation problems encountered in practice. This is clearly a judgement call. It is dangerous to extrapolate from the problems studied here to the set of all problems; however, this paper does show that in a diverse set of combinatorial optimisation problems there exist considerable similarities.

Where the similarities between problems exist, they are often qualitative rather than quantitative. This is true even for instances from the same class of problems. To obtain a clearer insight into the quantitative similarities, we have tried to study the scaling behaviour of the properties we measure as a function of the problem size. For Max-Sat and graph-colouring we can also show the change in behaviour as we vary the number of constraints (thus, exploring the change in the behaviour as we move through the phase transition). For TSP and quadratic assignment we can explore the behaviour as we vary the problem type. Thus, for example, in TSP we studied the behaviour of 11 different class of TSP [5]. Where appropriate, we show the results for these different classes of problem; however, to keep this paper readable, we have focused on Euclidean TSP. Many details for TSP and for the other problems have naturally been missed in this paper. Also some of the details of the analysis, such as the algorithm for finding local optima, and a full description of the use of principle component analysis has been omitted here, but can be found in the problem specific papers.

We have followed the same methodology in studying all the problems. Most significantly, we have used local search starting from many different random initial configurations to find the local optimal solutions. We have studied problems of size up to around 100 variables. For this size of problem, it has been possible to obtain a good representative sample of the local optima and in particular most, if not all, of the low cost local optima, which tend to have a much larger basin of attraction than average cost local optima. From these local optima we have computed different properties which we report in the paper.

### A. Previous Literature

The concept of fitness landscape goes back to Sewell Wright [11] who introduced it to understand biological evolution. Fitness landscapes have been studied for many years, both

in the physics community [12], [13], and in the optimisation community [14], [15], [16], [17]. In the evolutionary algorithm community, work on the fitness landscape started in the early 1990s [18], [19], [20], [2]. An early influential measure of the roughness of the fitness landscape was the autocorrelation [1], [21], [22]. Later the fitness-distance correlation was introduced to measure problem difficulty [2], [23], [24], [18]. Algebraic properties of the solutions in the landscape [16], [25], modality (number of local optima) [26] and the fractal dimension of the fitness landscape [27] are other examples. 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 [28], [29]. Fitness clouds is another method proposed to visualise the fitness landscape, which tries to represent some properties of the fitness landscape [30], [31], [32], reflecting the problem hardness.

During the last two decades many researchers have studied the landscape of optimisation problems including Travelling Salesman [33], [34], [35], Quadratic Assignment [23], Knapsack [36], [37], Max-Sat [38], [39], [3], graph drawing [40], Graph-Colouring [41], [42], [43], [44], [17], [45], evolutionary antenna design [46], flow-shop scheduling [47] and Bayesian network structure [48] problems.

The rest of this paper is organised as follows. In the next section we introduce the four combinatorial optimisation problems studied in this paper and describe the local search algorithm we use to find the optima in the landscape. Section III describes two bulk properties of the landscape, namely the auto-correlation and the mean fitness in a Hamming sphere around a given configuration. In section IV we present a study of the properties of global and local optima, including the number of steps a local-search algorithm takes to get to a local optimum, the number of local and global optima and distance between the optima, and the probability of reaching a local optima at a given cost. We also use principle component analysis to obtain a better understanding of the structure of the local optima. We draw conclusions including making comparisons between different problems in Section V.

## II. PROBLEMS AND METHODOLOGY

In this section, we describe the four different combinatorial optimisation problems studied in this paper and explain the way we generate the problem instances. We finish this section with a discussion about the local-search algorithms we use to find the local optima, and the way we distinguish different local optima from each other.

### A. Problem Definition

1) *Maximum Satisfiability Problem*: The Max-Sat problem is closely related to the satisfiability decision problem, colloquially known as SAT. This problem involves a set of Boolean variables  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  and a set of disjunctive clauses consisting of a subset of literals (a literal is either a variable or its negation). For example, a clause might be  $x_1 \vee \neg x_5 \vee x_{10}$ . Each clause can be considered as an additional constraint that we want to satisfy.  $K$ -Sat are instances of Sat where each clause consists of  $K$  literals. Max-Sat is the generalisation of

SAT to problems which are not fully satisfiable. It asks the question whether there exists an assignment of the variables which satisfies all but  $T$  clauses. Max- $K$ -Sat is NP-hard for  $K \geq 2$ . We will treat Max-Sat as an optimisation problem in which we try to find a configuration of variables which minimises the number of unsatisfied clauses. In this paper we consider Max-3-Sat. We use the objective function as the number of unsatisfied clauses, which we seek to minimise. Assuming there are  $m$  clauses and denoting the clauses by  $g_i(\mathbf{x})$ , then the cost function is given by

$$c(\mathbf{x}) = \sum_{i=1}^m \llbracket g_i(\mathbf{x}) \text{ is not satisfied} \rrbracket, \quad (1)$$

where  $\llbracket \textit{predicate} \rrbracket$  denotes the indicator function which is equal to 1 if the *predicate* is true and 0 otherwise.

Our focus will be on randomly generated instances, where each clause consists of  $K$  randomly chosen variables which are negated with probability of a half. We require each variable in a clause to be different and all clauses to be unique. We denote the number of variables by  $n$ , the number of clauses by  $m$ , and the ratio of clauses to variables by  $\alpha$ . In this paper we consider  $K = 3$ , for different values on  $n$  and  $\alpha$ . We note that Max-3-Sat has a phase transition for  $\alpha \approx 4.25n$  (the fact that this phase-transition scales linearly with  $n$ , makes it particularly easy to carry out a scaling analysis of this problem).

We look at the Hamming neighbourhood. The classic search algorithm for this neighbourhood is known as GSAT, which always chooses the best available neighbouring configuration. This can be efficiently implemented to run in  $O(1)$  time (see [3] for details).

2) *Graph-Colouring Problem*: The Graph-Colouring problem is a combinatorial optimisation problem which belongs to the class of NP-Hard problems. Given an undirected graph  $G(\mathcal{V}, \mathcal{E})$ , with a vertex (node) set  $\mathcal{V}$  and edge set  $\mathcal{E}$ , and  $k$  different colours, the Graph-Colouring problem is defined as finding a colouring of the vertices to minimise the number of edges whose vertices share the same colour. We denote a configuration of the Graph-Colouring problem with  $k$  colours as a vector  $\mathbf{x}$  of size  $n = |\mathcal{V}|$ , with elements  $x_i \in \{1, 2, \dots, k\}$  representing the colour of the  $i$ -th node. The cost of a configuration  $\mathbf{x}$  is defined as the number of *colour conflicts* in the graph, i.e., the number of edges whose vertices have identical colours,

$$c(G, \mathbf{x}) = \sum_{(i,j) \in \mathcal{E}} \llbracket x_i = x_j \rrbracket.$$

We consider the problem of finding a colouring,  $\mathbf{x}$ , which minimises the cost.

In this paper we will concentrate on instances drawn from the ensemble of random graphs  $\mathcal{G}(n, p)$ , consisting of graphs with  $n$  vertices where each edge is drawn with a probability  $p$ . In particular we focus on the case  $p = 0.5$ , so that the graphs are dense in the sense that typically the number of edges are of order  $n^2$ . For these instances we study the properties both as a function of the number of vertices  $n$  and as a function of the number of colours  $k$ .

The chromatic number,  $\chi(G)$  of a graph,  $G$ , is defined to be the smallest number of colours  $k$  such that a configuration

exists with no colour conflicts (i.e. a cost of zero). The chromatic number for most graphs in  $\mathcal{G}(n, p)$  is very heavily concentrated and differ by one at the most. The chromatic number marks the phase-transition between satisfiable and unsatisfiable problems. We will see, just as for Max-Sat, many other properties are determined by the position of this phase transition. Unfortunately, unlike in Max-Sat, the position of the phase transition does not depend linearly on the instance size; thus it is not easy to carry out a simple scaling analysis. Note that although an asymptotic result for the Chromatic number of random graphs is known [49], it provides a poor estimate for finite size graphs. A much tighter upper bound for finite size random graphs is given in [50]. In table I we give the chromatic number for different graph sizes. For a more complete discussion on the chromatic number see [4].

TABLE I  
MEAN CHROMATIC NUMBER FOR GRAPHS DRAWN FROM  $\mathcal{G}(n, 1/2)$ .

$n$	10	20	30	40	50	60	70	80	90	100
$\chi(n)$	3	5	6	7	8	9	10	11	12	13

3) *Travelling Salesman Problem*: The Travelling Salesman Problem (TSP) is arguably the most famous combinatorial optimisation problem. The classic description of the problem is that 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 once and then returns to the start city. Despite its deceptively simple description, it is surprisingly difficult to solve. In general, it is NP-hard, although for some problem types such as Euclidean TSP, there are polynomial time approximation schemes [51]. In this paper we present results mainly for Euclidean TSP, although where it is interesting, we present results for a number of different instances. These instances were proposed in [52], [53]; details can be found in [5]. For the Euclidean instances the cities are placed uniformly at random in a  $10^6 \times 10^6$  square.

The constraint that a solution to TSP must be a valid tour strongly restricts the move set. The classic move set for TSP is  $k$ -opt, where the tour is broken into  $k$  sections at  $k$  edges. These sections are then stitched back together to create a new tour. We have examined 2-opt, 3-opt and 4-opt; however, we predominantly consider 3-opt as we found 2-opt has so many local optima that it made the analysis computationally difficult. Furthermore, 2-opt always reverses part of a tour, making it a highly non-local operator for asymmetric problems. On the other hand 4-opt had such a huge neighbourhood that it was computationally challenging to ensure a local optimum is reached.

4) *Quadratic Assignment Problem*: The quadratic assignment problem is the least well-known combinatorial optimisation problem discussed in this paper. The problem is to find an assignment between a set of  $n$  facilities,  $\mathcal{F}$ , and a set of  $n$  locations,  $\mathcal{L}$ , which minimises the communication costs between facilities. More formally, we assign a flow between facilities  $w: \mathcal{F} \times \mathcal{F} \rightarrow \mathbb{R}$  and a distance between locations  $d: \mathcal{L} \times \mathcal{L} \rightarrow \mathbb{R}$ . Then the cost of an assignment  $\mathbf{x}: \mathcal{F} \rightarrow \mathcal{L}$

is taken to be

$$c(\mathbf{x}) = \sum_{i,j \in \mathcal{F}} w(i,j) d(x_i, x_j), \quad (2)$$

which is to be minimised. If all the distances were the same, then this reduces to the linear assignment problem which is solvable by the Hungarian algorithm in  $O(n^3)$  operations. However, with different distances this problem is in general NP-hard. Although we have described quadratic assignment in terms of facility and distances, just like TSP, it arises in many different guises in real applications.

An assignment can be viewed as a permutation; however, unlike TSP there is no further constraint. Thus the simplest move set is to swap the location of two randomly chosen facilities. To generate instances of this problem, we have to make two matrices; one is the distance matrix containing the distances between every pair of locations, and one the flow matrix containing the flow between every pair of facilities. To do so we randomly place the locations in a  $100 \times 100$  square and the flows between the pair of facilities are chosen uniformly randomly between 0 and 100.

### B. Methodology

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 operators. A neighbourhood is defined for each problem type. For the Max-3-Sat, Graph-Colouring and the Quadratic Assignment problems a simple Hamming neighbourhood is used. For the Travelling Salesman problem a 3-opt neighbourhood operator is used.

We also need to define the distance measure. This could be defined as the number of local search moves between two configurations or the number of non-common variables in two configurations. For Max-3-Sat we use the number of non-common variables which is also known as Hamming distance. For the TSP we take the number of non-common edges in two tours as the distance. Because of the symmetry in the landscape, for the case of the Graph-Colouring problem, two distance measures could be defined. One is the Hamming distance and the other is what we call the partition distance, which is defined as

$$D_p(\mathbf{x}, \mathbf{y}) = \min_{\pi} D_h(\mathbf{x}, \pi(\mathbf{y})), \quad (3)$$

where  $\pi(\cdot)$  is a permutation operator which permutes the colours. The minimisation is over all possible permutations of the  $k$  colours. The partition distance measures the smallest number of reallocations of partition membership to make the partition represented by  $\mathbf{x}$  into the partition represented by  $\mathbf{y}$ . When the Hamming distance is small, it is often the same as the partition distance.

In this paper, we define a local minimum to be a connected set of configurations with no lower-cost neighbours. One way to study the topology of a fitness landscape is to search for local optima by an exhaustive search of the landscape; however, this would confine us to studying small problems with sizes up to 20 or 30 variables (see, for example, [54]

and [55]). Instead, we use a local search algorithm which allows us to study larger instances. To find a local minimum we search all neighbours of the current configuration. If there is an improving move we take it, otherwise, if there are neighbours of the same cost, we systematically explore all their neighbourhoods, keeping a record of all configurations we have explored, and all configurations whose neighbours we have still to explore. This guarantees that either we find an improving move or we have reached a local optimum. Once we reach an optimum we use the lexicographic least member as a unique identifier of the minima (in graph colouring we permute the colours to a canonical ordering beforehand to ensure that all minima which are identical up to a colour permutation are treated identically). This approach limits the size of problem we can explore, either because the neighbourhood becomes too large (as happens in TSP using 4-opt) or because the size of the local optima is too large (as happens around the phase-transition in Max-Sat and graph colouring).

## III. BULK PROPERTIES

In this section we study two properties that are not directly related to the local optima. The first one is the auto-correlation function and the second is the mean fitness in a Hamming-sphere about a particular configuration. These measures tell us about the average ruggedness of the landscape and long-range correlations in the landscape.

### A. Auto-Correlation

The auto-correlation measures the local ruggedness of the fitness landscape by measuring the expected correlation of a random configuration and that of a configuration where we have made  $\tau$  randomly chosen steps [1]. For a problem instance, the auto-correlation is given by,

$$R(\tau) = \frac{1}{\sigma^2} \mathbb{E}((c(\mathbf{x}(t+\tau)) - \bar{c})(c(\mathbf{x}(t)) - \bar{c})),$$

where  $\mathbf{x}(t)$  is the configuration at step  $t$  of a random walk, and  $\sigma^2$  is the variance in the cost for random configurations of the problem instance. To compute this, we consider a random walk starting at an arbitrarily chosen initial configuration and move to a Hamming neighbour at each step.

The empirically measured log-auto-correlation for the Max-Sat, Graph-Colouring, Travelling Salesman and Quadratic Assignment problems for different instance sizes is represented in Figure 1. In all cases the fall off of the logarithm is almost linear, indicating an exponential fall off with a characteristic ‘‘correlation length’’ (note that the deviation for large  $\tau$  which is most prominent for TSP is most likely due to sampling errors). The correlation length is proportional to the system size for all the problems. We observe that in all cases the landscapes are fairly smooth, and become smoother as the system size increases.

For Max-Sat we show the auto-correlation for different ratios of clauses to variables,  $\alpha$ . For  $\alpha = 2$  almost all instances have a relatively large fraction of satisfiable configurations. In contrast, at  $\alpha = 10$ , almost no instances are satisfiable, with instances around  $\alpha = 4.3$  marking the phase-transition. The

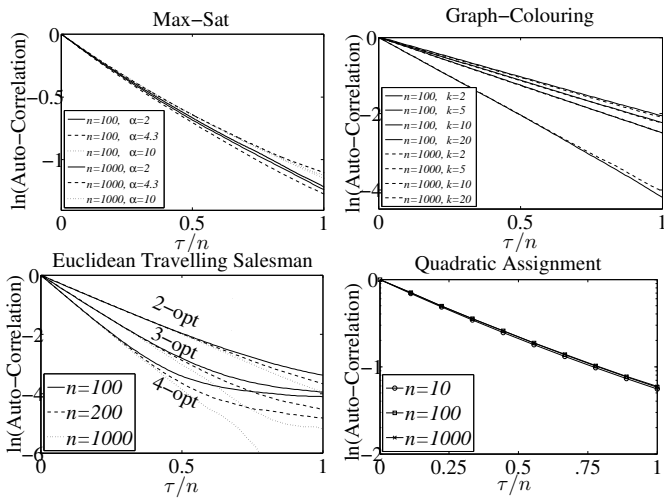


Fig. 1. The logarithm of the auto-correlation of the fitness landscape for the Max-Sat, the Graph-Colouring, the Travelling Salesman and the Quadratic Assignment problems for different values of  $n$  and parameters plotted against the time difference  $\tau/n$ . The results are for randomly generated problem instances. The number of steps is  $10^9$ .

behaviour of the auto-correlation gives no indications of this rather important change in the behaviour of the problem. For graph-colouring the correlation length grows as  $n(1 - 1/k)/2$  (see [56], [4]), where  $k$  is the number of colours. The phase-transition depends non-linearly on both  $k$  and  $n$ . As the difficulty of the problem instances varies depending on the proximity of the phase-transition, we observe that the auto-correlation function is not a good measure of problem difficulty. For the Euclidean-TSP we show the autocorrelation function for different search operators. As would be expected, the more disruptive the move operator the more quickly the correlation function falls.

### B. Expected Cost in Hamming Sphere

The auto-correlation shows the existence of a strong correlation in the cost of neighbouring configurations. Another measure which shows the same behaviour is the expected cost of a configuration in a Hamming Sphere from some reference configuration. This can often be computed analytically (see [3], [4]). We show plots of this, starting from different low-cost minima in figure 2. The costs fall off from their initial value to the average cost as we move towards the expected Hamming distance between randomly chosen configurations. This behaviour arises because the cost functions involve a sum of terms each of which depends on only a few variables. Thus, perturbing a few variables results in a relatively small change in the cost of a solution. One consequence of this behaviour is that low cost solutions have a long range influence on the fitness landscape. Furthermore, we will see later that the lower cost minima tend to have larger basins of attraction than the higher cost minima.

## IV. PROPERTIES OF LOCAL OPTIMA

In the rest of this paper we concentrate on properties of local optima. These were found using the exhaustive local

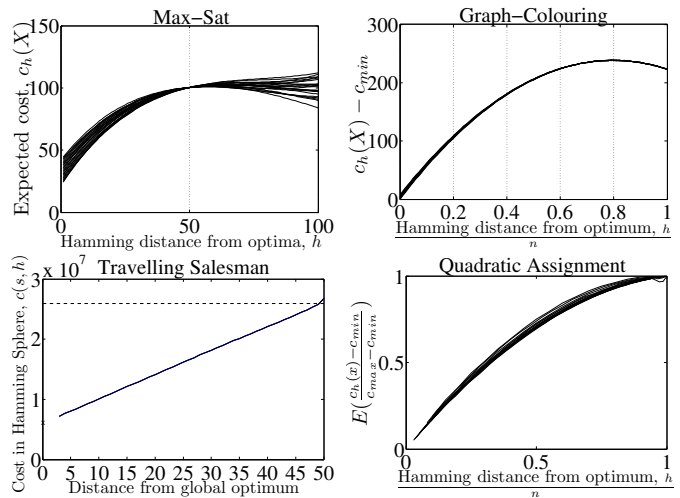


Fig. 2. Expected cost of the configurations in a distance sphere for the most local optima at each cost for different problems of different size. For the Max-Sat  $\alpha = 8$  and for the Graph-Colouring  $k = 5$ . The data are averaged over  $10^5$  different runs.

search, described in section II-B, many thousands of times to obtain a representative sample of local optima with large basins of attraction. As we argue later we believe that this sample includes almost all low cost solutions and in particular the global optima.

### A. Time to Local Optimum

We consider properties about the search process, starting with the time (number of moves) to reach an optimum. Note that we count the number of moves used, rather than the number of fitness evaluations, since in highly optimised local search algorithms data structures are used to avoid or speed up fitness evaluations. The time complexity of the local search is dominated in these algorithms by the update of the local search after making a move. We discuss this further at the end of this section. In this section, we distinguish between the constraint-satisfaction problems (Max-Sat and graph-colouring) and the other two. As discussed earlier, the constraint satisfaction problems undergo a phase-transition from an easy phase to a hard phase, as the number of constraints increases (assuming the number of variables is kept fixed). In figure 3 we show the empirically measured mean time to reach a local optimum versus  $\alpha$  for Max-Sat and versus  $k$  for graph-colouring.

For Max-Sat, small  $\alpha$  corresponds to the easy-phase with a small number of constraints. For small  $\alpha$  there is a relatively high proportion of the configurations that satisfy all the constraints. A local search algorithm will consequently find a globally optimal solution quickly. However, as  $\alpha$  increases towards the phase transition (which occurs around  $\alpha = 4.3$ ), local search slows down dramatically as the searcher explores very large plateau regions. Above the phase-transition the search plateau regions become less common and the search rather rapidly finds a local optimum. Exactly, the same pattern of behaviour is observed for graph-colouring. Note that for graph colouring we have examined changing the number of colours  $k$ . A large number of colours makes it easier to find

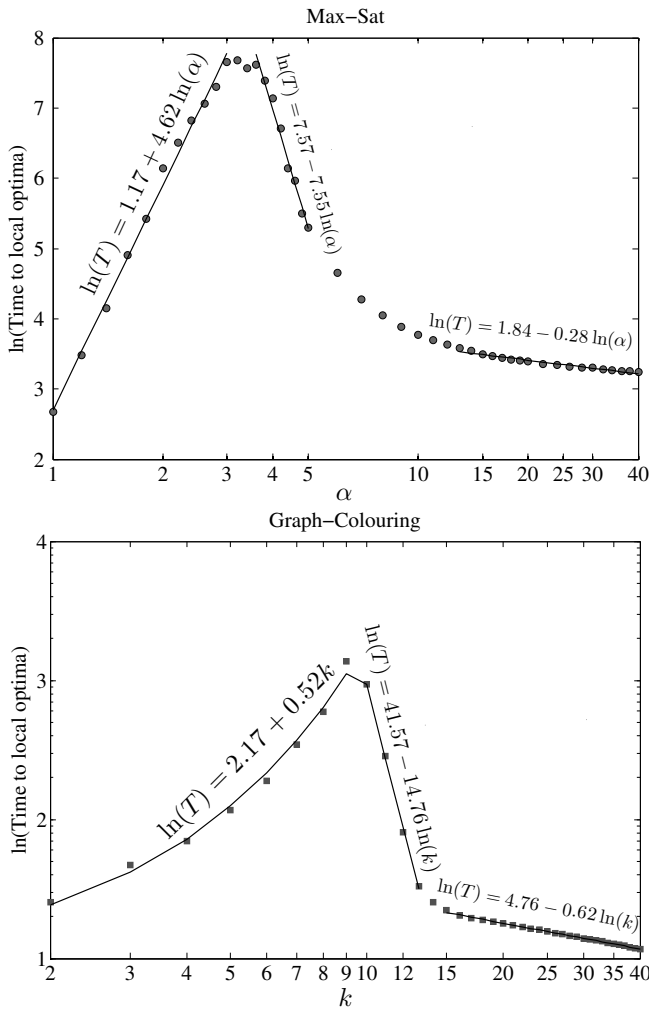


Fig. 3. Logarithm of the time to reach a local optimum for the problems with phase transition. The data for the Max-Sat problem are plotted versus  $\alpha$  on a log-log scale, where the size of the problem is  $n = 50$ . The data for the Graph-Colouring problem are plotted against the number of colours  $k$  on a logarithmic scale for  $n = 100$ . Each data point represents the mean over 100 instances and  $10^5$  hill-climbs per instance.

a satisfying solution; thus, the easy phase occurs on the right side of figure 3(b) and the hard side on the left (which is a mirror image of what occurs in figure 3(a) for Max-Sat). The phase-transition occurs for this size of instance at  $k = 9$ .

In the Max-Sat problem, graph representing time to local optima versus  $\alpha$  consists of three stages. From  $\alpha = 1$  up to  $\alpha = 3.2$  (phase transition), on a log-log scale, the time to local optima versus  $\alpha$  fits a straight line, suggesting that the time grows polynomially as  $T \approx \alpha^{4.16}$ . After the phase transition, there are the second and the third stages in which the time decreases polynomially. At the second stage, as  $\alpha$  increases, the time decreases rapidly by  $T \approx \alpha^{-7.55}$  (note that empirically it is difficult to distinguish between an exponential increase and a large polynomial—we fitted with a polynomial as it gives a slightly better fit). After reaching a certain point, the third stage begins, where the time decreases less rapidly as  $T \approx \alpha^{-0.28}$ .

A quite similar behaviour is seen in the Graph-Colouring problem. The time to reach a local optimum also consists

of three stages. On a logarithmic scale, at the first stage, from  $k = 2$  to  $k = 9$  (the chromatic number), the data fit a straight line, meaning that for  $k$  up to the chromatic number, the number of steps grows exponentially with the number of colours. The second and the third stages of the graph show different behaviours. On a log-log scale, the data for these two stages fit a straight line, suggesting that the time to local optima decays polynomially with the number of colours. Note that this time consists of the time taken on the flat regions, plus the number of improvement moves the local search algorithm sees during its search process. We will study each component separately in the rest of this paper.

The time to reach a local optimum as a function of the instance size is found to grow as a slow polynomial of the system size. This is shown in figure 4. For TSP we find a similar behaviour in a suit of instances (see [5]). The Max-Sat and graph-colouring data are for instances in the unsatisfiable (hard) phase.

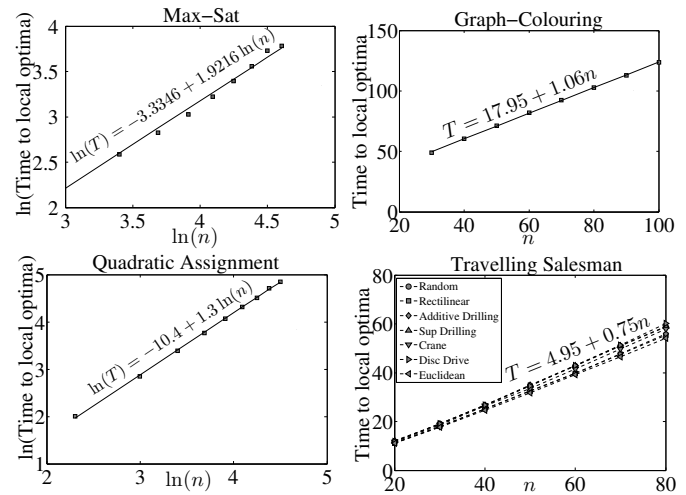


Fig. 4. Time to reach a local optimum for the problems versus the system size. For the Graph-Colouring problem  $k = 7$ , and for the Max-Sat problem  $\alpha = 8$ . Each data point represents the mean over 100 instances and  $10^5$  hill-climbs per instance.

We observe that the number of moves to reach the local optima grows sub-quadratically in all cases. To compare the run times between the different problems we also need to take into account the time taken to make a move. This depends on the implementation of the local search operator. The best algorithm known by the authors for solving Max-Sat can perform a move in  $O(1)$  operations, while that for graph-colouring takes  $O(n)$  [3]. TSP can be much slower, as it has a large neighbourhood ( $O(n^3)$  for 3-opt), and each move can require a non-local rearrangement of the tour. Efficient implementations of TSP usually restrict the neighbour and use sophisticated data structures to store the tour. The quadratic assignment problem is considerably simpler than TSP, but still has a large neighbourhood to explore  $(n(n-1)/2)$  compared to  $n$  for Max-Sat and  $(k-1)n$  for graph-colouring. Thus, the clock time can grow faster than figure 4 suggests. Nevertheless, the time taken to reach a local optimum is not a limiting constraint in finding a global optimum.

The analyses provided for different combinatorial problems

suggest that the time to local optima for these problems grows sub-quadratically with the system size. This, in a sense, means that in terms of the time to find a local optimum, the NP-Hard problems (or at least the ones studied in this research) become polynomially hard as the system size grows. However, this does not mean that finding the best solution to a NP-Hard problem can be performed in a polynomial time. There are other properties related to the hardness of the problems. The number of local optima and the correlation between the size of the basin of attraction of a local optimum and its cost are other important properties which are targeted in the next sections.

The average time to reach local optima hides details of how the search algorithms make progress. To study this we examine the change in cost as a function of the number of steps averaged over many different runs. This is shown in figure 5 as a density plot of the cost of a solution versus the number of steps (plotted on a logarithmic scale) for many different runs for the graph-colouring and the quadratic assignment problems. For graph-colouring (and similarly for Max-Sat) there is a characteristic dog-leg shaped curve, with rapid initial progress until the knee of the dog's leg is reached. At this stage on most runs a local optimum has been reached and no further progress is made, but on a small proportion of runs, large flat regions are reached and the rate of progress slows down significantly once this happens. The probability of reaching a large flat region increases considerably in the proximity of the phase-transition. In contrast, for quadratic assignment (and similarly for TSP), this second phase is not observed. This is a consequence of the fact that for quadratic assignment (and TSP) configurations with the same fitness are rare, so there are no large flat regions.

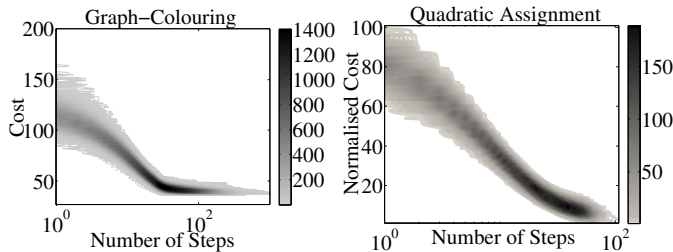


Fig. 5. The density plot of the record of the steps taken by the local search algorithm to get to the local optima for  $10^5$  different search process, starting from random configurations. The cost of the configurations at each step is represented against number of steps on a logarithmic scale. The size of the problem is  $n = 50$ . For the Graph-Colouring problem the number of colours is  $k = 5$ .

Note in figure 5 that we have plotted the number of steps on a logarithmic scale; thus, initially, the frequency of improving steps and the average size of an improving step is significantly larger than that which occurs later on in the search.

The time it takes for a local-search algorithm to find a local optimum consists of two components. One is the number of improving moves (cliffs) the search algorithm sees, and the other is the flat regions on which the algorithm wanders while searching for an improving move. These two properties change as the parameters of the problems change. Since there is no flat region in the landscape of the Quadratic Assignment and the Travelling Salesman problems, the time to local optima is

entirely made of improving moves. The landscape of Graph-Colouring and Max-Sat however, consists of both improving moves (when the search has reached a cliff edge) and flat regions. For the Graph-Colouring problem, we saw that up to the chromatic number, the time to local optima grows exponentially with the number of colours. Figure 6 shows the contribution of number of improving moves and the number of steps on flat regions in the time to local optima. Up to the chromatic number, the number of improving moves needed to reach a local optimum grows with the number of colours. After reaching its peak at the chromatic number, the number of improving moves decreases with the number of colours. The number of improving moves also depends on the system size. It seems that it grows linearly with the system size. On the other hand the number of steps on the flat regions has a more important role in the time to local optima. Quite similar to the behaviour we saw in time to local optima in figure 3, the graph representing the size of the flat regions in the landscape consists of three stages. At the first stage, from  $k = 2$  up to the chromatic number, the size of the flat regions grows exponentially with the number of colours. At the second stage, the size decreases polynomially with the number of colours until it reaches zero. And finally is the third stage at which the size of the flat regions is zero. This means that at the third stage (far above the chromatic number), the search algorithm directly reaches the local optima without wandering on the flat regions and the time to local optima consists entirely of the improving moves.

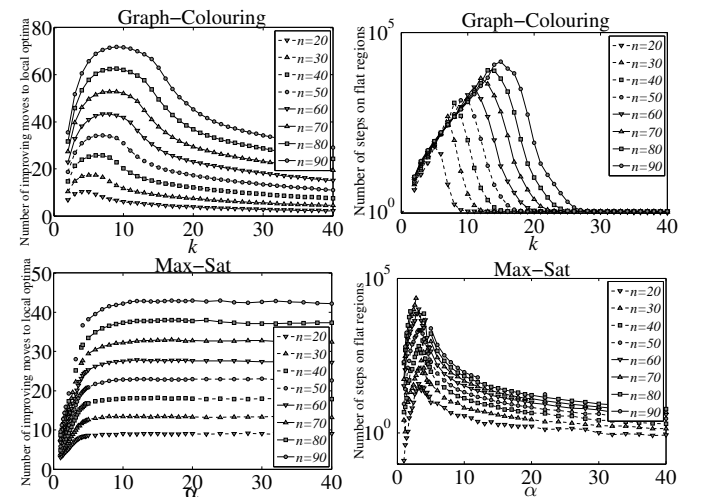


Fig. 6. Number of improvement moves and the size of the flat regions the local search algorithm sees during its search process versus the constraint parameter for different size of the problems. This is for the Graph-Colouring and the Max-Sat problems. Each data point represents the mean over 100 instances and  $10^5$  hill-climbs per instance. The data for the number of steps on the flat regions are plotted on a logarithmic scale.

A rather similar behaviour is seen in the Max-Sat problem. Up to the phase-transition, the number of improving moves need to reach a local optimum grows with the  $\alpha$ . Here, in contrast to the Graph-Colouring problem, after the phase-transition, the number of improving moves remains constant as  $\alpha$  grows. Figure 6 also shows the number of steps on the flat regions in the Max-Sat problem. Quite similar to the

Graph-Colouring problem, the size of the flat regions in the Max-Sat problem has three stages. From  $\alpha = 1$  up to the phase-transition, the size of the flat regions grows rapidly with  $\alpha$ . After the phase-transition, the size decreases polynomially with  $\alpha$ , but unlike the Graph-Colouring does not reach zero.

The analyses provided here show that unlike the Quadratic Assignment and the Travelling Salesman problems, which show no flat region in their landscape, for the problems with highly degenerative objective function, the size of the flat regions has a much more important role in the time to local optima than the number of improving moves has. This means that for these problems, most of the time of the local search algorithms is spent wandering on the flat regions (note that the data for the number of steps on the flat regions are plotted on a logarithmic scale). When wandering on the flat regions, it speeds up search if we do not check repetitive solutions. This property suggests that the Tabu search might be a more appropriate algorithm for the degenerative objective functions, particularly around their phase transition.

### B. Local Optima Properties

In this section, we explore properties of the local optima. As discussed above, these were found by performing multiple runs of the exhaustive search algorithm. As well as storing all the local optima, we count the number of times each optimum is visited. For all problems we found that fitter local optima were visited more often on average than less fit local optima. The mean number of visits for local optima binned according to their costs is shown in figure 7. For all four problems shown, the probability of visiting a local optimum appears to fall off exponentially with the cost. Similar behaviour was found for instances at other parameter values and also for different types of TSP (see [5]). The only exception was for TSP with random “distances”, where there was little, if any, correlation between the cost of the local optima and the probability of visiting them. We attribute this to the lack of significant long range structure in this problem.

Note that for low-cost (i.e. fit) local optima we visited each optimum a very large number of times. This provides some confidence that we have sampled almost all of these local optima (see the discussion on elves below for more on this). In contrast for high-cost local optima we often visited them only once, and we almost certainly have missed a large number of these. We are not aware of any means for estimating the number of such local minima. As we are significantly under sampling these, we cannot determine the probability of them being visited. Our best guess would be that the probability of visiting a local minimum continues to fall off exponentially with cost.

For local minima at the same cost, there is a considerable variation in the probability of them being visited. An explanation of this is due to the shadowing of minima by close by minima and the funnelling of the search into restricted parts of the search space. We cannot empirically rule out the existence of low cost (fit) solutions with extremely small basins of attraction. We have called such states elves (as they are virtually invisible entities, which we believe are unlikely to

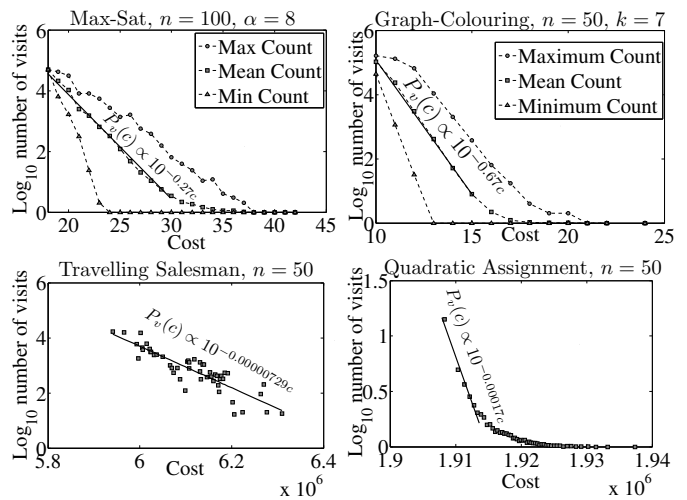


Fig. 7. Number of local optima are hit at each cost for different problems. For the Graph-Colouring and Max-Sat problems we show the maximum, mean and minimum number of times the local optima are hit at each cost level. For the Quadratic Assignment and the Travelling Salesman it is highly unlikely to have more than one local optimum at each cost, so the data show the number of times each local optimum is hit versus its cost. The size of the problem is  $n = 100$  for the Max-Sat problem and  $n = 50$  for the other problems. For the Max-Sat problem  $\alpha = 8$  and for the Graph-Colouring  $k = 7$ . The number of descents is  $10^6$ .

exist). In small instances where we can systematically explore the whole search space (see, for example, [54]), no elves were reported. Furthermore, we have seen no evidence of elves in any of problems we have investigated. From our discussion in section III-B, we know that low cost solutions depress the expected cost for all configurations that are more highly correlated than random configurations. Thus an elf cannot correspond to a deep and narrow well in the fitness landscape. For all these reasons we consider elves to be unlikely. In particular, we feel there is fairly strong evidence that for the size of problems we have examined, we have found all the global optima.

In figure 7, we see that the mean probability of visiting a local optimum at cost  $c$  is approximately exponential  $P_v(c) \propto 10^{-\lambda c}$ . The gradient,  $\lambda$ , depends on the properties of the instances. In figure 8 we show this gradient as a function of the system size and of  $\alpha$  for Max-Sat and  $k$  for graph-colouring. For Max-Sat this gradient is, to a first approximation, independent of the problem size, but quite dependent on the ratio of clauses to variables,  $\alpha$ . The situation is more complicated in graph-colouring, although as the position of the phase-transition does not scale linearly with  $n$ , this is not surprising. For both problems, as the number of unsatisfiable constraints increases, there is a small bias towards the low cost solutions.

Better local optima have, on average, exponentially greater chance of being found than less fit local optima, which clearly helps in finding a global optimum. However, there are exponentially more local optima at lower costs. We illustrate this in figure 9 where we show the proportion of local optima in different cost bins, and the probability of reaching a local optimum in a particular cost bin. We see clearly that the expected cost of a local optimum found by local search is



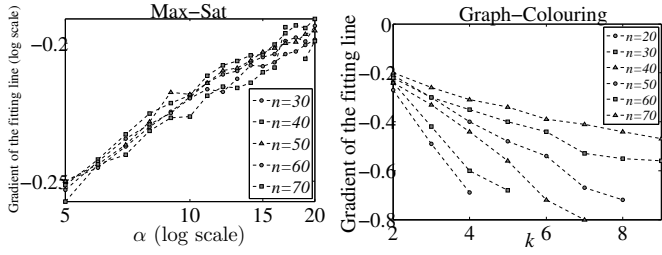


Fig. 8. The decay rate of the mean number of time the local optima at each cost level are hit (the gradient of the fitting line to the data in Figure 7) for different size of the problems versus the constraint parameters for the Max-Sat and the Graph-Colouring problems. The data are averaged over 50 randomly generated problem instances and  $10^6$  runs on each. The data for the Max-Sat problem are represented on a log-log scale.

substantially less than the mean cost of all local optima (note that we only show the local optima that have been found; we will almost certainly be underestimating the proportion of local optima on the right of each graph).

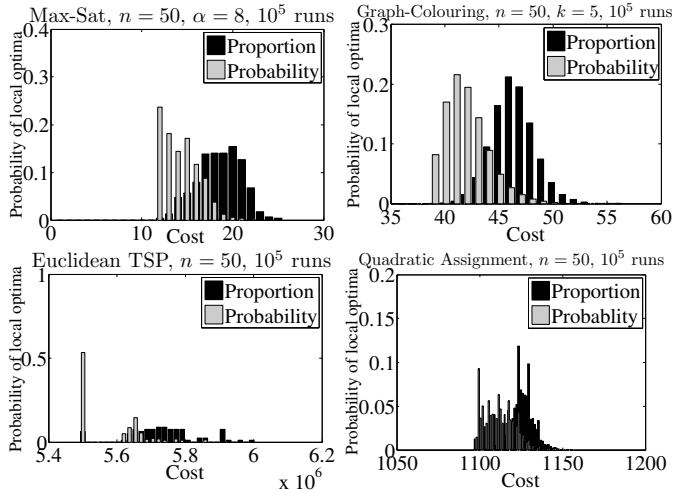


Fig. 9. Histograms showing the proportion of local optima at each cost value and the probability of finding a local optimum of each cost for one randomly generated instance of different problems. The size of the problem is  $n = 50$  for all the problems. The constraint parameters are set to  $k = 5$  and  $\alpha = 8$ .

Figure 9 suggests that there is a fairly high probability of finding a global optimum. This is true for moderate size problems of the type investigated here. However, the number of local optima grows rapidly with the instance size. We will see later that this results in the probability of local search finding a global optimum decreasing exponentially. Figure 10 shows the scaling behaviour of the number of local optima at different costs. We show the logarithm of the number of local optima in each cost bin, divided by some power of  $n$  chosen so that the peak of the curves are roughly similar. For quadratic assignment the vast majority of local optima were only seen once. It was thus not possible to estimate the scaling behaviour for the number of local optima using this method.

The curves indicate that the number of local optima at each cost grows roughly as

$$N(c) \approx e^{n^a g(c/c_v)},$$

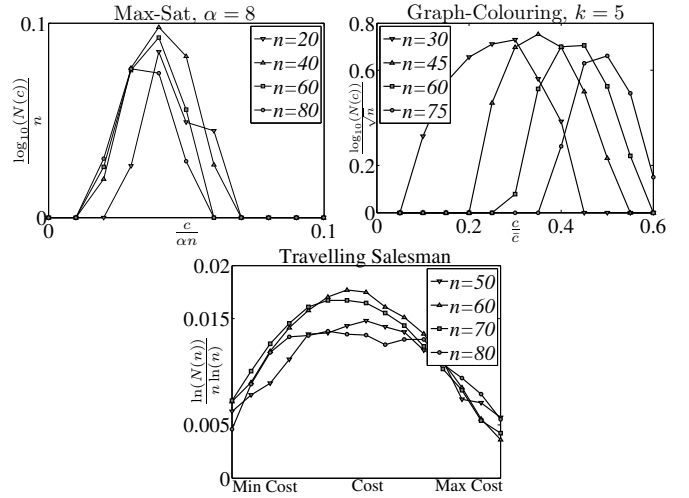


Fig. 10. Number of local optima at each cost scaled by the system size,  $n$  versus the cost divided by the average cost for different problems. This is averaged over 100 randomly generated problem instances for  $10^5$  runs.

where  $g(\cdot)$  is some approximately parabolic function (graph-colouring is the slight outlier as it is difficult to find a scaling relation due to the fact that the chromatic number does not scale linearly with  $n$ ). Each problem has a very different scaling behaviour with  $n$ . This is seen even within problem classes. Different types of TSP have different scaling behaviour. Note that for graph-colouring we count local optima that differ by a permutation symmetry as the same local optimum. Thus with the Hamming neighbourhood, due to the permutation symmetry, there are in fact  $k!$  more local optima. To estimate the growth in the number of local optima for quadratic assignment we measured the proportion of randomly sampled configurations which were local optima. This was only practical for problems up to size 12. From these very small instance sizes we estimated that the number of local optima grows approximately as  $\exp(0.63n - 2.2)$ .

The number of local optima also depends on the other details of the instances. In figure 11, we show the total number of local optima for Max-Sat versus  $\alpha$  and  $n$ , and for graph-colouring versus  $k$  and  $n$ . Note that because of the simple scaling behaviour of Max-Sat, we have scaled the y axis to remove the main variation in  $n$ . We observe that the number of local optima grows linearly (and rather slowly) as we increase the number of constraints  $\alpha n$ . The scaling behaviour of graph-colouring is more complicated and we just show the raw number of local optima as a function of  $n$  and  $k$  (note that for large  $n$  and  $k$  we will be substantially under sampling the number of local optima, so the rate of growth may be significantly greater than shown). The behaviour is more complicated than that for Max-Sat, since increasing  $k$  both increases the size of the search space (which grows as  $k^n$ ), and makes it easier to satisfy the constraints. These two effects tend to balance, resulting in the number of local optima remaining fairly constant for a considerable proportion of the graph.

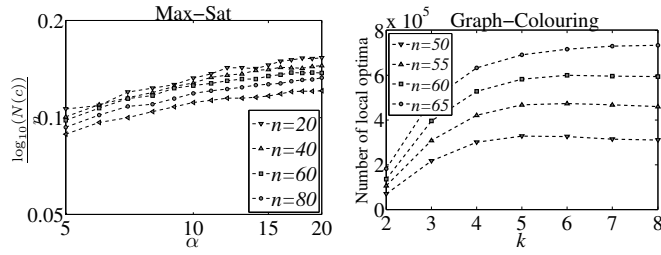


Fig. 11. The number of local optima versus the number of colours  $k$  and  $\alpha$  for the Graph-Colouring and the Max-Sat problem. The data are averaged over 100 randomly generated problem instances of different size. The number of runs is  $10^6$ . The data for the Max-Sat are plotted on a log-log scale.

### C. Global Optima

As the instance size increases, the gap between the cost of typical configurations found by local search and the global minimum cost grows. This is shown in the top two graphs in figure 12 for Max-Sat and graph-colouring. The y-axis has been scaled by  $n$  and the average cost ( $n^2$ ) respectively, so these gaps look linear. By plotting the Max-Sat result versus  $1/n$ , it is possible to extrapolate back for large  $n$ . The typical gap in this limit is around  $0.03n$ . The extrapolation for graph-colouring is more complicated. A plausible fit to the data available would be

$$c_{min} = \bar{c} \left( 1 - \frac{0.22}{\sqrt{n}} + O\left(\frac{1}{n}\right) \right),$$

where  $\bar{c} \approx n^2/(2k)$  is the average cost. What is clear is that the gap between the minimum cost and the average cost found by a local search is of order  $n^2$ .

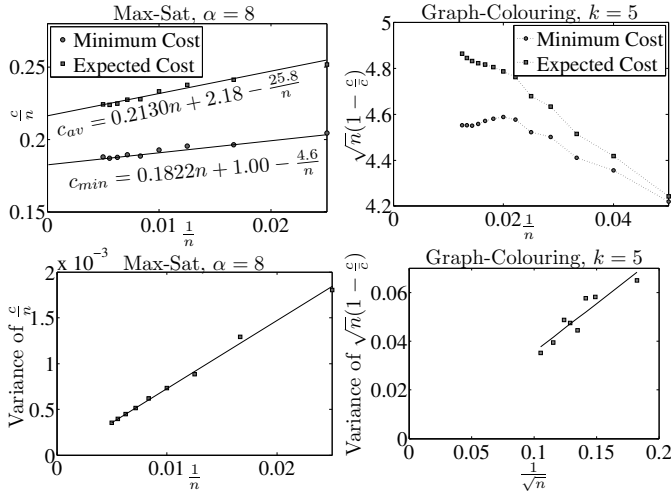


Fig. 12. Plot of the expected maximum fitness (expected minimum cost for Graph-Colouring), the average fitness (average cost for Graph-Colouring) found by local search algorithm and the variance in the fitness of the local optima versus  $1/n$  ( $1/\sqrt{n}$  for Graph-Colouring). This is calculated by performing  $10^5$  hill-climbs on 100 randomly generated instances of the problem.

The bottom graphs in figure 12 show the variance in  $c/n$  for Max-Sat and  $c/n^2$  for graph-colouring. We see that this variance decreases towards zero in both cases, indicating that as  $n$  increases, it becomes less likely for local search to find local minima with a cost close to the global minimum cost.

In figure 13, we plot the log-probability of local search finding a global minimum as a function of the instance size. The straight line fits are consistent with an exponential fall off in the probability of finding a local optimum for all these problems. Note that we have chosen parameters so that both Max-Sat and graph-colouring are in their unsatisfiable (hard) phase. The exponents indicate the rate at which the problems become difficult (at least for the local search we have used). By this measure the quadratic assignment problem becomes harder, substantially faster than the other problems, followed by graph-colouring.

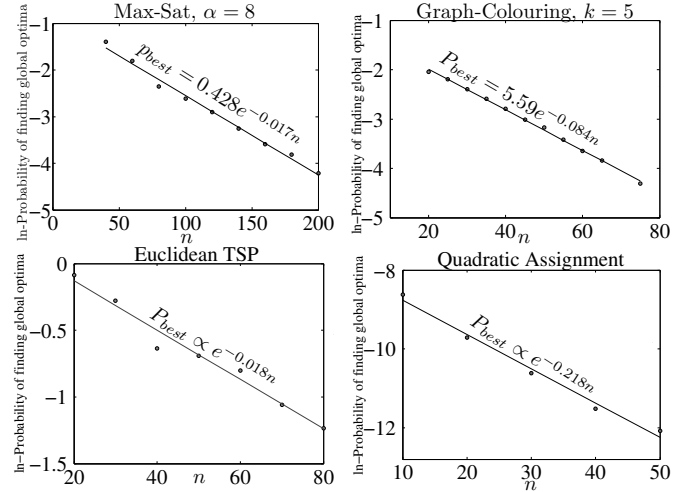


Fig. 13. Natural log-probability of finding the maximum fitness optima versus  $n$  averaged over 100 randomly generated instances. The number of runs is  $10^6$ . For the Graph-Colouring problem the number of colours is  $k = 5$  and for the Max-Sat problem  $\alpha = 8$ .

For Euclidean TSP and quadratic assignment the probability of two configurations having the same cost is very low and consequently there will nearly always be a unique global optimum. This is not the case for Max-Sat and graph-colouring where there is only a relatively small number of cost levels. In figure 14 we show a histogram of the number of local optima. In just over one third of cases the global optimum is unique, but in a reasonable number of cases there can be quite a few global optima.

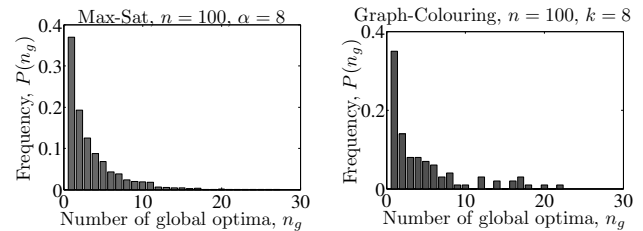


Fig. 14. Histogram of the number of global optima for 10 000 random Max-Sat and Graph-Colouring instances with  $n = 100$  where  $\alpha = 8$  and  $k = 8$ . The number of runs is  $10^5$  on each problem instance.

Often the global optima are quite close to each other in Hamming distance, but on occasion uncorrelated global optima can exist. This is shown in figure 15 where we have calculated the Hamming distance between global optima

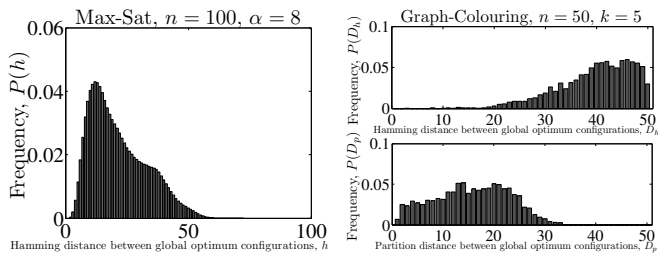


Fig. 15. Histogram of Hamming distances between the global optima in the Max-Sat and Graph-Colouring problems. For the Graph-Colouring problem we have also included the histogram of partition distances. The histograms are made from 10 000 randomly generated problem instances. For the Max-Sat problem  $n = 100$  and  $\alpha = 8$ , for the Graph-Colouring  $n = 50$  and the number of colours is  $k = 5$ .

averaged over a large number of instances with multiple global optima. The mean distance between randomly chosen pairs of configurations for Max-Sat with  $n = 100$  is 50. For graph-colouring we show both the Hamming distance and the partition distance between global optima (the partition distance being the minimum Hamming distance between two configurations up to a permutation in the colours). The expected Hamming distance between random pairs of configurations for this graph-colouring problem instance is 40, while the mean partition distance is around 33. In both problems we see that there can exist global optima which are unrelated to one another.

#### D. Proximity to Global Optima

Given that it is exponentially unlikely to find a global optimum, an important question is whether it is possible to infer information about the position of the global optimum (or a high quality optimum) from information provided by local optima. To study this, in figure 16, we show a density plot of the number of local optima as a function of the cost and the distance from the nearest global optimum. For graph-colouring we use the partition distance, since otherwise the colour-permutation symmetry would hide the fitness distance correlation. For TSP we show this both for Euclidean instances and for random instances. In all cases, there is some fitness distance correlation, although this is rather minimal for quadratic assignment and for TSP with random instances.

As fitter members of the population tend to be closer to a global optimum, this could allow population-based algorithms to search the landscape more effectively than local search. In Max-Sat an effective search method is to perform a number of independent searches and then to move to the centroid of the best solutions found by local search [39], [3] (that is, the solution with the lowest mean distance to the best solutions). Although the centroid is usually at a higher cost than any of the solutions found by local search, when the search is restarted, the cost rapidly decreases and typically beats any of the solutions found by local search. We can understand this behaviour from figure 16. Local search finds optima which tend to be at lower costs than the average optima. These are slightly correlated with the global optima. The centroid, although not itself an optimum, is correlated with the global optimum. Although this method does not typically find the

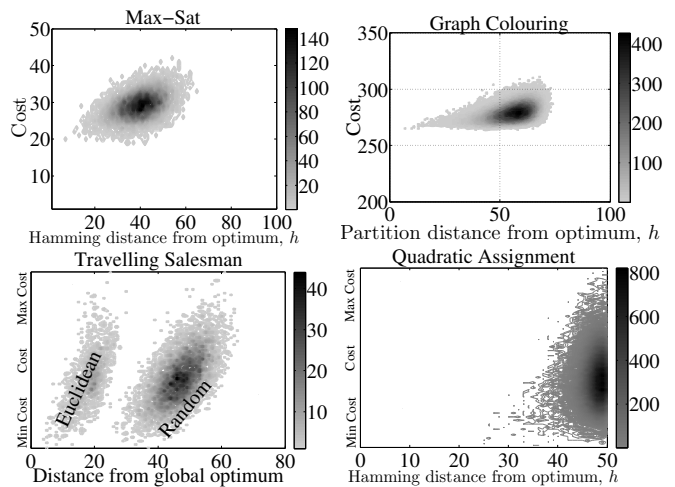


Fig. 16. Density plot of local optimum configurations as a function of their cost and Hamming distance from the most frequently visited global optimum. For the Max-Sat and the Quadratic Assignment problems we use the Hamming distance; for the Graph-Colouring problem the partition distance and for the Travelling Salesman problem the number of non-common edges are used as the distance measure. The number of colours in the Graph-Colouring problem is  $k = 5$  and in Max-Sat  $\alpha = 8$ . For the Travelling Salesman problem two types of the Random and the Euclidean are used. The number of runs is  $10^6$ .

global optimum solution, it will often find a solution close to a global optimum with a low cost.

In graph-colouring, one of the most effective algorithms over the last 10 years for solving dense random graphs is a hybrid genetic algorithm [57], [4]. Again we can understand this in terms of the density plot shown in figure 16. This algorithm uses a specially tailored crossover which finds a child with a small partition distance between its two parents. The crossover is hybridised with a local search. The local search finds solutions which are below average cost and thus tend to be correlated with a global optimal solution. Crossover produces new configurations which are no longer optimal, but tend to be more correlated with a global optimum. Reapplying local search tends to find a nearby above-average solution, which is likely to be even more correlated with the global solution.

However, the fitness distance correlation is rather weak. In figure 17 we show the average distance to the global optima for configurations in different cost bins. The axes have been rescaled to make the data fall on top of each other for different problem sizes (we were unable to do this for quadratic assignment).

For both Max-Sat and graph-colouring we find that the mean distance to a global optimum appears to scale approximately as

$$d_{nearest} \approx n \left( \frac{c - c_{min}}{n} \right)^{0.33}.$$

Therefore the mean distance to the global optimum for a local optimum with cost of one greater than the global optimum scales as  $n^{0.67}$ . The number of configurations in a Hamming-ball of size  $h$  is given by  $\binom{n}{h} \geq (n/h)^h$  so that the number of configurations in a Hamming ball with a radius of  $d_{nearest}$  is greater than  $n^{0.33n^{0.66}}$ . Within this radius there are no lower cost

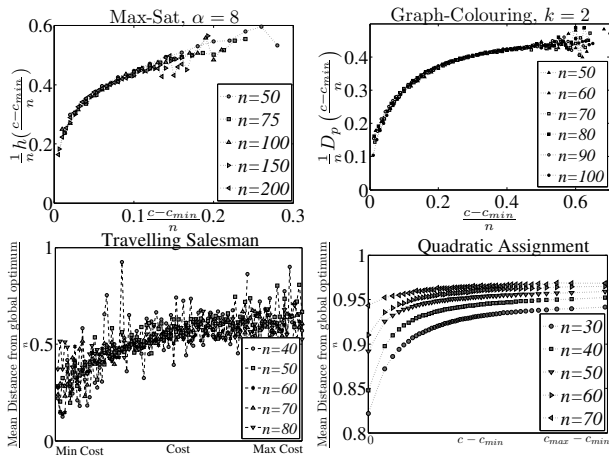


Fig. 17. Measure of the mean distance to the closest global optimum from a local optimum of cost  $c$  for different problems of different size. The data are averaged over 100 problem instances and the number of runs is  $10^6$ . For the Max-Sat problem  $\alpha = 8$ , and for the Graph-Colouring the number of colours is  $k = 2$ .

configurations providing heuristic information about where to search. The situation for TSP is no better, with very fit solutions having in expectation around  $0.2n$  edges different from the global solution. The number of configurations in a Hamming ball of this size is  $\binom{n}{0.2n} (0.2n)! = n! / (0.8n)! \approx n^{0.2n}$  (although not all these configurations are valid tours). The situation is even worse for quadratic assignment where the gap between a good solution and the global optimal solution is greater than  $0.8n$  and the number of configurations in a Hamming ball of this size is  $\binom{n}{0.8n} (0.8n)! = n! / (0.2n)! \approx n^{0.8n}$ . Thus, finding a global optimum, even when starting from the next fittest local optimum, is still extremely challenging.

### E. Principle Component Analysis

The local optima are correlated; however, a single measure of the correlation provides very little information about the directions in which the local optima differ. The very large number of these local optima may actually just vary in a relatively small number of ways. For example, in TSP we can imagine a situation where there are many small local alterations that are independent of each other. This can give rise to an exponential number of local optima by choosing different combinations of local variations. To explore the variation in local optima, we perform principle component analysis (PCA) on the set of local optima. For Max-Sat we can compute PCA directly on the binary vectors representing the solutions. For graph-colouring we consider vectors where each component represents a pair of edges in the graph. We assign 1 to the component if the edges are connected to nodes of the same colour and zero otherwise. Clearly, this representation is invariant to colour permutations. For TSP we again consider a representation in terms of the occupancy of each possible edge in the tour. Finally, for quadratic assignment we consider the occupancy of an edge in the bipartite graph between facilities and locations.

This approach to exploring the set of local optima was first introduced in [5]. That paper shows that each eigenvector can

be viewed as showing possible alternatives between different ways of solving TSP. It was also shown that, by projecting the data in the space spanned by the first few principle components, it was often possible to find clustering in the data that was not transparent otherwise. Often the high quality optima were confined to a single cluster, suggesting that this approach could potentially be used for guiding search.

In this paper we consider only the spectrum of eigenvalues, that is the values of eigenvalues. Note that the expected reconstruction error of a local minimum is proportional to the sum of the eigenvalues that are dropped. Thus this spectrum provides a measure of the number of relevant directions in the space of local optima. In figure 18 we show the eigenvalue spectrum for the four problems of interest.

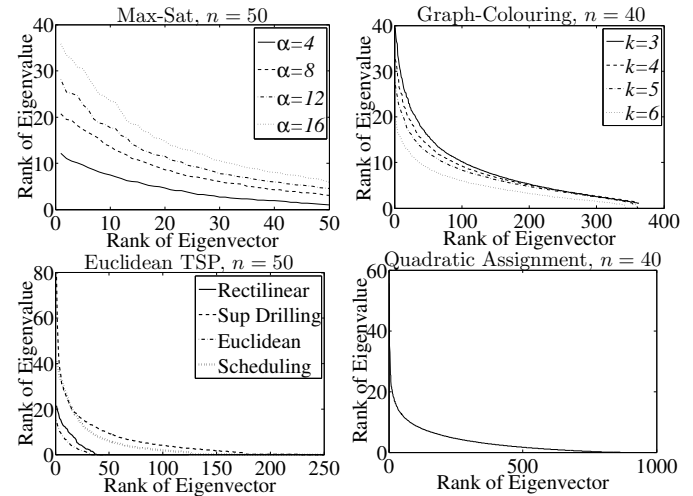


Fig. 18. Plot of the eigenvalues of the covariance matrix of local optima versus their rank.

Note that the size of the eigenvalues measures the number of components in which the solutions differ from each other. A rapid fall off shows that the variation is dominated by only a few major directions, while a slower fall off shows that local optima can be constructed in very different ways. Max-Sat has by far the smallest number of directions in which the data can differ. This reflects the much smaller search space. Note that in Max-Sat the variation in the local optima increases considerably with  $\alpha$ . This shows that increasing  $\alpha$  considerably increases the size of the space in which the local optima sit. All the other problems have a very rapid initial drop off in the eigenvalue spectrum, indicating that there is a small number of directions where there are very prominent changes in the solutions. In graph-colouring we see that increasing  $k$  decreases the size of the space spanned by the local optima, despite increasing the size of the search space. In TSP we see that different problem types lead to very different eigenvalue structures. Euclidean TSP is somewhat atypical in being particularly easy to solve. This is reflected in that the eigenvalue spectrum falls off faster than that of other problems. Finally, we note that quadratic assignment has a large number of non-zero eigenvalues, which may in part explain its difficulty (although note that graph-colouring with  $k = 3$  shows slightly more variation).

## V. CONCLUSION

This paper provides a comparison between four very different combinatorial optimisation problems. We have had to be selective in the data we present; a more complete set of data is available in the problem specific papers [3], [4], [5], [6]. The first observation is the remarkable degree of similarity we observe in the four problems. Some of this is qualitative only, and appears more similar than it really is, due to selection of the scale of the axes. Nevertheless, there are substantial similarities. As we argued earlier, we believe the cause of these similarities is that the structure of the cost function is similar in so far as they are all sums of a large number of largely independent components (clearly they are not completely independent as this is what makes the problems hard). Furthermore, changing a small number of variables only affects a small number of components, so there is a long range correlation between costs. This results in a multi-modal fitness landscape with some correlation between local minima.

At least for the properties we examined (which we believe are all relevant to designing or choosing a search algorithm), this similarity seems to be much more important than the differences in the problem. These differences include the neighbourhood structure, the size of the search space and symmetries. In designing algorithms, these features are essential at an operator level (i.e. designing mutation or crossover operators for EAs). However, at a high level (e.g. using a population to explore the search space over long distances), these differences seem much less important and the variation within a problem class can be as significant as variations between problem classes. Being able to separate the low-level (short-scale) properties from the large-scale properties is clearly of the utmost importance in developing meta-heuristics.

Despite the similarities, there are some important differences between problems. Most significantly, in the problems we have studied, is the division between the constraint optimisation problems with a satisfiability phase transition and TSP and quadratic assignment which do not show the phase transition. Although all four problems behave similarly, when Max-Sat and graph-colouring are above the phase transition, there is a significant qualitative difference around and below the phase transition. With a small number of constraints (i.e. significantly below the phase transition) Max-Sat and graph-colouring have a significant proportion of configurations which satisfy all the constraints, and they tend to be easy to solve. Around the phase transition, there can be exponentially large low-cost flat regions which can dominate the search time. At the same time, some of these correspond to local rather than global optima. Efficient methods for navigating through this space, such as Tabu search or using additional heuristic information, can substantially speed up search for these problems.

None of the behaviour we observe is that surprising, although there are properties that would be difficult to guess and occasionally contradict commonly held beliefs. We outline some of these below.

- Auto-correlation is often used as a measure of problem difficulty, and yet, as we have seen, all four hard problems

we have studied have a low correlation length. Furthermore the correlation length is oblivious to properties such as the phase transition where the problem difficulty changes drastically. In our opinion the importance of auto-correlation is often over played in studies of fitness landscapes.

- In all four problems the presence of a local optimum strongly influences the expected cost of a configuration up to a Hamming distance equal to the random Hamming distance between configurations. One consequence of this is that fitter local optima tend to have significantly larger basins of attraction than less fit local optima as their whole neighbourhood is on average fitter than that of a less fit local optimum.
- The time for local search to reach a local optima is poorly studied, although clearly of the utmost importance in designing heuristic search algorithms. The number of moves taken to reach an optimum grows sub-quadratically as a function of the problem size for all four problems investigated. However, it can grow exponentially large as we approach the phase transition (for those problems with a phase transition). The cause of this is the large amount of time spent on very large flat regions of the search space. The position of the phase transition is the most important determiner of the behaviour of the problem for constraint satisfaction problems. In the phase where it is not possible to satisfy many of the constraints, the constraint satisfaction problems (Max-Sat and graph colouring) exhibit similar behaviour to TSP and quadratic assignment.
- In all problems investigated, the number of local optima grows exponentially as a small polynomial of the instance size. This measure in combination with the basins of attraction determines the difficulty of finding a global optimum.
- The probability of finding a global optimum falls off exponentially with the instance size; however, this fall off is often sufficiently slow that medium sized problems (often up to several hundred elements) can be solved quite efficiently by running local search multiple times. Interestingly, these sized problems are often found in benchmarking libraries. Extreme caution is required in extrapolating results obtained on such problems as repeated local search becomes ineffective when considering larger problems.
- In all problems very fit local optima often are a considerable distance apart. This is particularly evident in Max-Sat and graph colouring where frequently there are multiple global optima. These will often have small correlations with each other showing that it is often possible to obtain optimal solutions in very different ways.
- Despite the above observation there is a small but significant positive correlation between fitness and the proximity to a global optimum. The structure of this, however, can vary considerably between problem types. An evolutionary algorithm using selection and crossover (possibly hybridised with a local search algorithm) may be able to exploit this correlation.

- In all problems the mean distance from a global optimum to the next best optimum is sufficiently large that finding the global optimum by exhaustively searching the proximity of the next best optima is intractable in problems of any size.
- Finally, although the number of local optima increases exponentially in some small polynomial of the system size, there are often only a relatively small number of ways in which the local optima vary. This is revealed by the rapid fall off in the size of the principle components.

Optimisation is not easy. There are many properties that can affect the choice of search algorithm to be used. We have explored a number of them in this paper and a few more properties in the problem specific papers [3], [4], [5], [6]. However, there is a surprising, but pleasing uniformity in the behaviour of a lot of apparently different optimisation problems. This holds out the promise that the task of understanding fitness landscapes, for at least a large group of real world problems, is not endless. We believe there is the hope of modelling many of the properties of a fitness landscape for many different problems using only a small number of parameters; hence, providing a powerful classification system for landscapes. We have not attempted that in this paper, but rather, provided the raw data, as we believe it may be of use to other researchers interested in understanding the anatomy of fitness landscapes.

## REFERENCES

- [1] E. D. Weinberger, "Correlated and uncorrelated fitness landscapes and how to tell the difference," *Biol. Cyber.*, vol. 63, pp. 325–336, 1990.
- [2] T. Jones, "Evolutionary algorithms, fitness landscapes and search," Ph.D. dissertation, University of New Mexico, Albuquerque, 1995.
- [3] A. Prügel-Bennett and M.-H. Tayarani-N., "Maximum satisfiability: Anatomy of the fitness landscape for a hard combinatorial optimisation problem," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 3, pp. 319–338, 2011.
- [4] M. H. Tayarani and A. P. Bennett, "Anatomy of the fitness landscape for graph-colouring problem," *submitted to Journal of Evolutionary Computation*, MIT press, vol. 0, p. 1, 2012.
- [5] M. H. Tayarani and A. Prügel Bennett, "Travelling salesman problem: A landscape analysis," *submitted to Journal of Evolutionary Computation*, MIT press, vol. 10, p. 1, 2012.
- [6] M. H. Tayarani and A. P. Bennett, "Quadratic assignment problem: A landscape analysis," *submitted to IEEE Transaction on Systems, MAN, and Cybernetics Part B: Cybernetics*, vol. xx, p. 1, 2012.
- [7] D. Sherrington and S. Kirkpatrick, "Infinite-ranged models of spin-glasses," *Physical Review: B*, vol. 17, no. 11, pp. 4384–4403, 1978.
- [8] E. Gardner, "The space of interactions in neural network models," *J. Phys. A*, vol. 21, p. 257, 1988.
- [9] H. Horner, "Dynamics of learning for the binary perceptron problem," *Zeitschrift für Physik: B*, vol. 86, pp. 291–308, 1992.
- [10] B. Derrida, R. B. Griffiths, and A. Prügel-Bennett, "Finite-size effects and bounds for perceptron models," *Journal of Physics: A*, vol. 24, pp. 4907–4936, 1991.
- [11] S. Wright, "The roles of mutation, inbreeding, crossbreeding, and selection in evolution," in *Proceedings of 6th Congress of Genetics*, vol. 1. ACM Press, 1932, p. 365.
- [12] M. Mézard, G. Parisi, and M. A. Virasoro, *Spin-Glass Theory and Beyond*, ser. Lecture Notes in Physics. Singapore: World Scientific, 1987, vol. 9.
- [13] A. K. Hartmann and M. Weight, *Phase Transitions in Combinatorial Optimization Problems*. Wiley-Vch, 2005.
- [14] B. A. Huberman and H. T., "Phase transitions in artificial intelligence systems," *Artificial Intelligence*, vol. 33, no. 2, pp. 155–171, 1987. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0004370287900336>
- [15] P. Cheeseman, R. Kanefsky, and W. M. Taylor, "Where the really hard problems are," in *Proceedings of the 12th international joint conference on Artificial intelligence - Volume 1*, ser. IJCAI'91. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1991, pp. 331–337. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1631171.1631221>
- [16] L. K. Grover, "Local search and the local structure of NP-complete problems," *Operations Research Letters*, vol. 12, pp. 235–243, 1992.
- [17] A. Hertz, B. Jaumard, and M. P. de Aragao, "Local optima topology for k-coloring problem," *Discrete Applied Mathematics*, vol. 49, pp. 257–280, 1994.
- [18] B. Manderick, M. de Weger, and P. Spiessens, "The genetic algorithm and the structure of the fitness landscape," in *Proceedings of 4th International Conference on Genetic Algorithms*, 1991, pp. 143–150.
- [19] K. Mathias and D. Whitley, "Genetic operators, the fitness landscape and the traveling salesman problem," in *Parallel Problem Solving from Nature*. Elsevier Science Publishers, 1992, pp. 219–228.
- [20] S. Forrest and M. Mitchell, "What makes a problem hard for a genetic algorithm? some anomalous results and their explanation," Tech. Rep. 3, 1993.
- [21] E. Angel and V. Zissimopoulos, "Autocorrelation coefficient for the graph bipartitioning problem," *Theoretical Computer Science*, vol. 191, pp. 229–243, 1998.
- [22] J. Czogalla, "Fitness landscape analysis for the continuous flow-shop scheduling problem," in *Proceedings of 3rd European Workshop, Evo*, Naples, 2008.
- [23] P. Merz and B. Freisleben, "Fitness landscape analysis and memetic algorithms for the quadratic assignment problem," *Evolutionary Computation, IEEE Transactions on*, vol. 4, no. 4, pp. 337 – 352, Nov. 2000.
- [24] R. Lefticaru and F. Ipate, "A comparative landscape analysis of fitness functions for search-based testing," in *IEEE 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, USA, 2008.
- [25] P. Stadler, "Towards a theory of landscapes," *Complex Systems and Binary Networks*, pp. 78–163, 1995.
- [26] J. Horn and D. E. Goldberg, "Genetic algorithm difficulty and the modality of fitness landscapes," *Foundations of Genetic Algorithms*, vol. 3, 1995.
- [27] T. Hoshino, D. Mitsumoto, and T. Nagano, "Fractal fitness landscape and loss of robustness in evolutionary robot navigation," *Auton. Robots*, vol. 5, pp. 199–213, 1998. [Online]. Available: <http://dx.doi.org/10.1023/A:1008874222544>
- [28] S. Verel, P. Collard, M. Tomassini, and L. Vanneschi, "Fitness landscape of the cellular automata majority problem: View from the "olympus"," *Theor. Comput. Sci.*, vol. 378, pp. 54–77, 2007. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1243519.1243807>
- [29] S. Vérel, P. Collard, M. Tomassini, and L. Vanneschi, "Neutral fitness landscape in the cellular automata majority problem," *CoRR*, vol. abs/0803.4240, 2008.
- [30] P. Collard, S. Vérel, and M. Clergue, "Local search heuristics: Fitness cloud versus fitness landscape," *CoRR*, vol. abs/0709.4010, 2007.
- [31] G. Lu, J. Li, and X. Yao, "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*, ser. EvoCOP'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 108–117.
- [32] L. Vanneschi, M. Tomassini, P. Collard, S. Vrel, Y. Pirola, and G. Mauri, "A comprehensive view of fitness landscapes with neutrality and fitness clouds," in *Genetic Programming*, ser. Lecture Notes in Computer Science. Springer Berlin - Heidelberg, 2007, vol. 4445, pp. 241–250.
- [33] K. Mathias and D. Whitley, "Genetic operators, the fitness landscape and the traveling salesman problem," in *Parallel Problem Solving from Nature*. Elsevier Science Publishers, 1992, pp. 219–228.
- [34] P. F. Stadler and W. Schnabl, "The landscape of the traveling salesman problem," *Physics Letters A*, vol. 161, no. 4, pp. 337 – 344, 1992. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0375960192905573>
- [35] K. D. Boese, "Cost versus distance in the travelling salesman problem," UCLA computer science department, Los Angeles, Tech. Rep., 1995.
- [36] H. Yoshizawa and S. Hashimoto, "Landscape analyses and global search of knapsack problems," in *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, vol. 3, 2000, pp. 2311 –2315 vol.3.
- [37] J. Tavares, F. B. Pereira, , and E. Costa, "Multidimensional Knapsack Problem: A Fitness Landscape Analysis," *IEEE Transactions on Systems, Man, and Cybernetics -Part B*, vol. 38, no. 3, pp. 604–616, 2008.
- [38] Weixiong and Zhang, "Configuration landscape analysis and backbone guided local search.: Part i: Satisfiability and maximum satisfiability," *Artificial Intelligence*, vol.

- 158, no. 1, pp. 1 – 26, 2004. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0004370204000542>
- [39] M. Qasem and A. Prügel-Bennett, "Learning the large-scale structure of the max-sat landscape using populations," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 4, pp. 518–529, 2010.
- [40] R. Lehn and P. Kuntz, "A contribution to the study of the fitness landscape for a graph drawing problem," in *Applications of Evolutionary Computing*, ser. Lecture Notes in Computer Science, E. Boers, Ed. Springer Berlin / Heidelberg, 2001, vol. 2037, pp. 172–181. [Online]. Available: [http://dx.doi.org/10.1007/3-540-45365-2\\_18](http://dx.doi.org/10.1007/3-540-45365-2_18)
- [41] A. Hertz, B. Jaumard, and M. P. de Aragão, "Local optima topology for the k-coloring problem," *Discrete Appl. Math.*, vol. 49, pp. 257–280, 1994.
- [42] J. P. Hamiez and J. K. Hao, "An analysis of solution properties of the graph coloring problem," in *4th metaheuristics international conference*, Porto, Portugal, 2001.
- [43] J. Culberson and I. Gent, "Frozen development in graph coloring," *Theor. Comput. Sci.*, vol. 265, pp. 227–264, August 2001.
- [44] H. Bouziri, K. Mellouli, and E. G. Talbi, "Fitness Landscape Analysis for Optimum Multiuser Detection Problem," *Journal of Combinatorial Optimization*, vol. 21, no. 3, pp. 306–329, 2009.
- [45] H. Bouziri, K. Mellouli, and E.-G. Talbi, "The k-coloring fitness landscape," *Journal of Combinatorial Optimization*, vol. 21, pp. 306–329, 2011. [Online]. Available: <http://dx.doi.org/10.1007/s10878-009-9249-2>
- [46] J. Alander, L. Zinchenko, and S. Sorokin, "Analysis of fitness landscape properties for evolutionary antenna design," in *Artificial Intelligence Systems, 2002. (ICAIS 2002). 2002 IEEE International Conference on*, 2002, pp. 363 – 368.
- [47] J. Czogalla and A. Fink, "Fitness landscape analysis for the no-wait flow-shop scheduling problem," *Journal of Heuristics*, pp. 1–27. [Online]. Available: <http://dx.doi.org/10.1007/s10732-010-9155-x>
- [48] Y. Wu, J. McCall, and D. Corne, "Fitness landscape analysis of bayesian network structure learning," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*, June 2011, pp. 981 –988.
- [49] B. B., "The chromatic number of random graphs," *Combinatorica*, vol. 8, no. 1, pp. 49–55, 1988.
- [50] A. Prügel-Bennett, "On the chromatic number of finite random graphs, journal ="
- [51] S. Arora, "Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems," *J. ACM*, vol. 45, no. 5, pp. 753–782, Sep. 1998. [Online]. Available: <http://doi.acm.org/10.1145/290179.290180>
- [52] J. Cirasella, D. S. Johnson, L. A. McGeoch, and W. Zhang, "The asymmetric traveling salesman problem: Algorithms, instance generators, and tests," in *Revised Papers from the Third International Workshop on Algorithm Engineering and Experimentation*, ser. ALENEX '01. Springer-Verlag, 2001, pp. 32–59.
- [53] R. L. Rardin, C. Tovey, M. Pilcher, and P. Pardalos, "Analysis of a random cut test instance generator for the tsp," in *World Scientific*, 1993.
- [54] J. Hallam and A. Prügel-Bennett, "Large barrier trees for studying search," *IEEE Transaction on Evolutionary Computation*, vol. 9, no. 4, pp. 385–397, 2005.
- [55] W. Benfold, J. Hallam, and A. Prügel-Bennett, "Optimal parameters for search using a barrier tree Markov model," *Theoretical Computer Science*, vol. 386, 2007.
- [56] P. Stadler, "Landscapes and their correlation functions," *Journal of Mathematical Chemistry*, vol. 20, no. 1, pp. 1–45, 1996.
- [57] P. Galinier and J. K. Hao, "Hybrid evolutionary algorithms for graph coloring," *Journal of Combinatorial Optimization*, vol. 3, no. 4, pp. 379–397, 1999.



and bioinformatics.

**Adam Prügel-Bennett** received the B.S. degree in physics from the University of Southampton, Southampton, U.K., in 1984, and the Ph.D. degree in theoretical physics from the University of Edinburgh, Edinburgh, U.K., in 1989.

He worked in research jobs in Oxford, Paris, Manchester, Copenhagen, and Dresden before finally returning to the University of Southampton. He is currently a Reader in the Department of Electronics and Computer Science. His research interests include evolutionary algorithms, machine learning,



**Mohammad-H. Tayarani-N.** is currently a Ph.D. student at the University of Southampton, Southampton, U.K. His main research interests include evolutionary algorithms, machine learning, and fractal image compression.