Reaching summits is not wandering or Getting insight into problem landscapes to go higher, faster

Philippe Preux and Denis Robilliard and Cyril Fonlupt
Laboratoire d'Informatique du Littoral
BP 719, 62228 Calais Cedex, France
Philippe.Preux@lil.univ-littoral.fr
Tel: (+33).321.970.046
and El-Ghazali Talbi and Vincent Bachelet
Université de Lille 1
Laboratoire d'Informatique Fondamentale de Lille
59655 Villeneuve d'Ascq Cedex, France

January 15, 1999

Category: GA

Abstract

Basically, this paper says that if you want to reach the top of a mountain, you should keep it in focus, which is obvious. It also says that the topology of mountains is not the same at their top than in the valleys, which is something also obvious. It says that going from the valley up to a certain altitude is rather easy and within the ability of most people, but going higher is more difficult and a matter of practice, again something known by everyone. So, this paper says a few obvious things and applies these trivialties to combinatorial optimization by way of GAs. In the case of combinatorial optimization, these evidences seem to be less evident, or, at least, are sometimes forgotten. It also shows that considering these facts and keeping them in mind while designing a search algorithm, one can go pretty high with rather little effort.

1 Introduction

The term "landscape" has been repeatedly invoked these last years in the EC community though seldomly being used to improve search algorithms. A landscape is defined in the context of iterated local search algorithm by its operator that define the neighborhood of a point, the fitness function, for the instance under consideration (we restrict the definition here to operators that transform one point into one other point). Then, let us define the graph $\mathcal{G} = (\mathcal{E}, \mathcal{N})$ where the set of nodes $\mathcal{N}$ is composed of the points of the search space; each node is linked to all of its neighbors for the given operator. Finally, an altitude is associated to each point, this altitude being related to the fitness of the point. In principle, edges are directed and have a strength that indicates the probability to obtain the point at the end from the point at the other end of the edge. Since Kauffman's papers on Nk-landscapes [14, 15], we are used to the idea of valleys, plateaus, peaks and massif centrals among which search algorithms travel, seeking the highest or the deepest points. Kauffman convincingly argued that the simple variation of the parameter k completely modifies the structure of the landscape. From a single peaked landscape for $k = 0$, the best points come to be concentrated into a tiny region of the space (so-called massif central) when k increases but remains close to 1, and progressively they diffuse within the landscape, finally turning into a rugged structure with a large number of peaks being spread in the whole landscape for $k = N - 1$. In the same time, the number of peaks (that is, local optima) increases from 1 in the whole space made of 2^N points for $k = 0$ to 1 point out of N points for $k = N - 1$, while the average quality of local optima decreases as k increases [30].

For combinatorial optimization problem (COMOPT), Weinberger [29] has proposed to study the distribution of fitness of points visited along a random walk. Starting from a point taken at random, a random walk iteratively selects a neighbor of the current point to become the new current point. Then, Stadler and Schnabl showed for the symetric Traveling Salesman Problem (sTSP) that the auto-correlation of the series consisting of these fitnesses is $e^{-2/n}$ for the 2-change operator, $e^{-4/n}$ for the city swap operator [26]. They argue that a higher auto-correlation is likely to be a sign of a smoothier ladnscape, thus the sign of a more efficient operator. This provides some ground to the well-known observed superiority of 2-change on city-swap for the sTSP. Manderick *et al.* have measured the correlation between the finess of a point and its child by the mutation operator of a GA (and between the fitness of a couple of points and their children by a recombination operator). They have found a high correlation between this measure and the observed efficiency of the operators when the GA solves the TSP [19]. Jones has also contributed to these ideas, though not tackling COMOPT [12].

However interesting, these works conceal a major weakness when one is interested in optimization. Stadler and Schnabl uses an averaging approach to obtain their results, thus describing average points of the search space. Both Manderick *et al.* and Jones measure the performance of operators on random points, that is average points of the search space in the case of COMOPT. So, strictly speaking, this all deals with regions of the landscape that contain average points; however, when optimizing, we are not interested in average points which can be very easily obtained by random sampling, but we are interested in better than average points. Thus, we are interested in a description of these points, of the region(s) they lie in, and the/a way to reach them starting from random points. So, averaging approaches actually describe a very large part of the landscapes[1] which contains no point significantly better than the average. Indeed, averaging approaches describe $100 - \epsilon$ % of the landscape, but this is this very tiny ϵ which is of interest when optimizing, thus when using GAs for optimization or other iterated local search algorithms such as hill-climbers or tabu search.

Very naturally, Stadler and Schnabl have concluded that the landscape of the sTSP is isotropic thanks to their result on a constant auto-correlation along random walks. However, we also have good reasons to think that the sTSP landscape contains a massif central (see Kirkpatrick and Toulouse's work about the presence of a "deep valley" [16], Boese's [6], and [9]) which means than non average points (including local optima) are not spread at random in the whole landscape but concentrated in even tinier regions that those evoked previously (let us say a part $\eta \ll \epsilon$ of the whole search space). In some sense, this region is very specific in the landscape, surely different from the average $100 - \epsilon$ % region in some ways and using some relevant measure. However exceptionally tiny, this region yet contains much more points than can be enumerated by current top of the art enumeration algorithms such as branch-and-bound and branch-and-cut. However, it is this region that have to be investigated because it contains the optimal points in (rather) high concentration. The way to reach this region from a random point is very simple and efficiently done by a hill-climber. Furthermore, being different from the rest of the space, the operators being efficient in this part of the space can become useless in the η region, and conversely. For

[1] this explains the good agreement between Stadler and Schnabl theoretical work and Manderick *et al.* experimental work

an obvious example of this fact, 2-change is very efficient at finding the edge of the η region. But, once a local optima located on this edge has been reached, its efficiency becomes null and some tricks have to be used (*e.g.* heating, that is blind short jumps in simulated annealing, tabu list in a tabu search to eschew cycling), or an other operator can be called at rescue in order to reach better points. So, this means that different operators can be efficient in different parts of one landscape[2]. This comes in good agreement with the well-known fact that hybrid algorithms are generally more efficient than pure algorithms for COMOPT.

In the sequel of this paper, we first investigate the structure of the sTSP and particularly concentrate on the non average part of the landscape. We also use the results of this investigation to design a very efficient hybrid GA which finds very fastly high quality points by suitably combining rather simple operators that are acting when they are the most useful. Our results are slightly worse than those obtained with the TSP top of the art Johnson's ILK algorithm if comparison is made with regards to quality and time of execution. Then, we present our work on the structure of the Quadratic Assignment Problem which shows different kinds of landscapes ranging from one massif central, more than one massif centrals, to rugged landscapes. We also very briefly discuss the case of the simple Job-Shop-Scheduling Problem landscape. Finally, we draw some conclusions.

2 The Traveling Salesman Problem

We have investigated the landscape of the 2 dimensional sTSP [9]. We summarize here the methodology as well as the conclusions of this work.

To begin with, we remind that the TSP is an $\mathcal{NP}$-hard problem [10, 17]. Basically, a set of n points is located in a k-dimensional space. One has to find a permutation π that specifies the order in which the cities are visited in order to minimize the tour length, e.g., find π that minimizes $\sum_{i=1}^{i=n} d_{i,\pi(i)}$, where d_{ij} is the distance between cities i and j. Though this problem can be defined without any limitations on the metric of the space and k, we will consider the case of 2 dimensional TSP, with a symetric metric (basically, euclidian space). The TSP was considered in our study because it is a very well-known $\mathcal{NP}$-hard problem, very intuitive, for which we have very efficient search algorithms (see [17, 1] for a thorough and up-to-date reviews), and large libraries of benchmarks with which algorithms can be compared.

A very simple though efficient operator is the so-called 2-change which removes two edges from a tour and subsequently rebuild another one; there is only one choice for the rebuild. Starting from this idea, we generalize with 3-change, and k-change which respectively remove 3, k, edges and rebuild a new tour. For these operators, more than one single tour may be build. Finally, there exists the very efficient, and not so immediate to implement soundly, lk-change [18].

We will implicitly use a notion of distance between two tours that indicates the number of applications of the operator to transform one tour into the other one. However, we are not able to compute this distance excalty for 2-change. So, we approximate it by the number of edges that the two tours have in common. The soundness of this approximation has been discussed by Boese [6]. So, in the sequel, when we say that two tours are close to each others, we implicitly relate ourselves to this definition of distance.

An adaptive walk is an iterative algorithm that starts from one point in the search space, uses an operator to obtain its neighborhood and selects one better point in the neighborhood to become the next current point. The process iterates as long as a better neighbor exists. We stress the fact that the selection does not take the best neighbor, but simply a better neighbor. So, an adaptive walk is not a steepest ascent walk, though a constantly ascending walk. Furthermore, it does not need to be deterministic. More technically, an adaptive walk is very fast to run, much faster than steepest descent that have to search the

[2] which is going further than the idea that one operator is dedicated to a certain class of landscape, *ie* one operator = one landscape

neighborhood thoroughly to find the best neighbor.

Using 2-change, such an adaptive walk is able to reach good local optima in the case of the TSP. So, we perform thousands of adaptive walks to gather a collection of local optima. Then, we do some statistical analysis of the local optima that have been obtained as well as on the trajectory of the adaptive walks. This study have been published in [9] so that we remind the conclusions here very shortly:

- the local optima are gathered in a tiny region, centered around the shortest tour,
- the basins of attraction are highly intertwined so that two adaptive walks starting with two different tours typically end up on two different local optima,
- the local optima are rather close in quality from the shortest tour (10 % or less),
- the walks are $3n$ steps, starting from random tours, where n is the size of the instance.

We have designed a rather simple hybrid GA (see fig. 1). Basically, we know that a short tour is to be found within a valley. Thus, the idea is to first find a valley with mere adaptive walk, and then work inside the valley by recombining the local optima that have been found. At this stage, we can iterate the whole process to progress further inside the valley. We end-up with some kind of hybrid genetic algorithm. However, we have the advantage over standard approaches (see [22] for a review) to begin recombination in an area of the search space where it is really useful, instead of recombining tours of very low quality. This speeds-up our hybrid GA a lot as will be seen below.

Our approach has much to be compared with Martin *et al*'s Chained Local Optimization (CLO) [21, 20] and Johnson's Iterated LK [11]. In the next section, we first present our hybrid GA, briefly review Johnson's version of ILK and Jünger *et al*'s ILK, and provide some experimental results comparing the 3 algorithms. Note has to be taken that Johnson's ILK algorithm, together with Martin's CLO, are the best local search algorithms for the TSP to date, according to Johnson's review [11].

We use the Tsplib [24] benchmark library to assess our algorithm. We use populations of 10 individuals. Before all, the initial generation is optimized with 3-change adaptive walks before attempting recombination. New individuals are then created by crossover using a slightly refined version of the well-known enhanced-edge recombination (EER). We don't recall here the details of EER, and refer the reader to [31]. Our modification to EER relies on the fact that edges not belonging to any parents may have to be introduced in order to satisfy the tour constraint. It has been noted that this fact tend to decrease much the fitness of children. To avoid this, we use a ranked selection to choose a suitable new edge with a probability inversely proportional to its length. Eventually the children are optimized with a 3-change adaptive walk. We use a "malthusian" replacement for the population : firstly we delete doublons, to keep diversity as the population size is small, then parents and children are sorted by fitness and the best individuals form the new population. We have chosen to limit the number of generations in such a way that the number of 3-change adaptive walks is equal to the instance size, in order to compare our results with Johnson's ILK performances given in [11]. Table 1 shows our best results among 10 runs.

We compare our algorithm to the following:

- multi-start 3-change adaptive walks which runs a large number of adaptive walks based on the 3-change operator starting from tours sampled at random (column 1 of table 1, figures taken from [13]),
- multi-start lk-change walks restarting from slightly perturbed local optima (column 3, figures taken from [13]),
- Johnson's iterated lk-change (column 4, figures taken from [11]),
- EAX, a highly efficient algorithm to recombine tours (column 5, taken from [23]). To recombine 2 tours, EAX build cycles from edges com-

```
take a set of n points at random
do
    perform a 3-change adaptive walk on each current points
    recombine local optima
    remove duplicates to have a new set of n points
while some criterion is fulfilled
```

Figure 1: Pseudo-code of our hybrid algorithm.

ing from both parents, then connect them as well as possible using some local optimization.

The following observations can be made:

- from comparison of multi-start 3-change adaptive walks and our algorithm, we very clearly see that the recombination we use is useful and increases very much the quality of local optima that are found,
- from comparison of Jünger *et al*'s multi-start lk-change and our results, we see very clearly that the use of lk-change is not worth its cost of execution and design. Clearly, using the very simple to design 3-change and edge recombination yields higher quality of local optima, while the execution time is much lower for our algorithm as far as can be judged from [13],
- the comparison between our algorithm and Johnson's ILK shows that the latter provides higher quality local optima than our algorithm. However, this observation should be tempered by the fact that Johnson's algorithm is the results of 20 years of polishing while our algorithm is implemented in a rather crude fashion. Johnson's algorithm is also a bit faster than ours,
- the comparison between our algorithm and EAX algorithm [23] is also in favor of EAX. However, the computational requirements for EAX have nothing in common with those used by our algorithm: we use 10 individuals instead of many hundreds, and for a given instance, all our runs are done in the time of one EAX execution. Furthermore, EAX is strongly dedicated to TSP while our approach is more problem independent.

3 The Quadratic Assignment Problem

We have begun to apply the same analysis to the quadratic assignment problem (QAP).

The QAP is also an $\mathcal{NP}$-hard problem in which n objects between which there exists certain flows have to be placed in order to minimize the costs of exchanges. More formally, given two matrices, the flow matrix $\mathcal{F} = (f_{ij})$ and the distance matrix $\mathcal{D} = (d_{ij})$, we minimize $\sum_{ij} f_{ij} d_{\pi(i)\pi(j)}$, where π is a permutation of n integers.

In [2], we have used the concept of flow dominance introduced by Vollamn and Buffa [28] to help decide which algorithm has to be used to optimize the most efficiently a given instance. The algorithms that have been considered were tabu search, genetic algorithm without local search, and hybrid GA including local search.

The flow dominance is defined by $d_f = \frac{\sigma(f_{ij})}{<f_{ij}>}$ where $< . >$ denotes the average. Somehow, the flow dominance indicates whether there are some objects that dominate the others, that is, objects which placement contribute a lot in the resulting fitness of the permutation while the other objects are almost neglectable.

More recently [3], we have exhibited different structures for different instances of the QAP. In cunjunction with the flow dominance of an instance, we use the following statistics:

- the diversity in terms of solutions (also called

Instance	(1)	(2)	(3)	(4)	(5)
lin105	0.0	0.0	0.0		
pr107	0.0	2.05	0.0		
pr124	0.0	1.15	0.0		
pr136	**0.01**	6.14	0.38		
pr144	0.0	0.39	0.0		
pr152	0.0	1.85	0.0		
u159	0.0	11.49	0.0		
rat195	**0.34**	3.01	0.47		
d198	**0.0**	6.12	0.16		
pr226	0.0	1.72	0.0		
gil262	**0.08**	3.07	0.55		
pr264	**0.0**	6.04	0.49		
pr299	**0.01**	4.37	0.15		
lin318	**0.41**	2.67	0.53	0.14	*
rd400	**0.17**	3.42	0.75		
pr439	**0.07**	3.61	0.38		
pcb442	**0.28**	3.01	0.90	0.09	*
d493	**0.30**	3.32	0.84		
u574	0.72	4.61	0.60		*
rat575	**0.81**	4.46	1.03		*
p654	**0.02**	0.62	0.03		*
d657	1.04	3.52	0.74		*
rat783	**0.79**	4.22	0.91		*
pr1002	**0.68**	3.80	1.51	0.27	
pcb1173	2.11	5.26	1.46		*
rl1304	**0.47**	7.08	1.62		
rl1379	1.41	3.65	1.13		
u1432	1.08	5.39	0.99		
pr2392	2.20	5.26	1.75	0.16	

Table 1: On a set of instances of the Tsplib, we provide the quality of the best local optimum found by our algorithm in column (1), the results of a multi-start 3-change adaptive walk in column (2), the result of Jünger's muti-start lk-change in column (3), the results of Johnson's ILK in column (4). In column (5), an asterik * indicates the instances for which the optimal solution is found by EAX. Differences between results and the optimum are expressed in %. Bold-faced results are those for which our algorithm does better than Jünger's ILK.

the entropy) of local optima reached by adaptive walks,

- the diversity in terms of fitness of these local optima,
- the gap between these diversities for local optima and random points,
- the number of steps performed by adaptive walks.

On a set of benchmarks issued from the QAPLib [7], we have found the following structures:

- local optima are concentrated in one massif central (like the sTSP),
- local optima are gathered in different massif centrals,
- local optima are spread in the whole search space.

As long as local optima are relatively well gathered, it is rather easy to find good local optima. When they come to be spread in the whole landscape, it becomes very difficult to find a local optimum of very good quality. The walks are short, leading to poor quality local optima.

When the local optima are gathered in more than one massif central, the idea is to first split the set of local optima into different patches that correspond to the different massif centrals. Then, using the same kind of technique based on a carefully recombinating operator as for the TSP is expected to yield good results. Benchmarks are under work.

4 Conclusion and perspectives

In this paper, we have argued for a study of the structure of landscapes of problems to improve search algorithm; we have also argued for the study of non average points of landscapes, average points being non interesting in the sense of optimization. Measures of auto-correlation along random walks are not enough to characterize landscapes in non average region. In the case of TSP, auto-correlation can be the same in landscapes having a very small massif central, or in a landscape having larger ones. So, clearly, we need other means to characterize landscapes.

However, using the fact that there exists a high concentration of local optima in the sTSP, we have designed a very efficient algorithm based on the iterative recombination of local optima, and their subsequent local optimization. Even if our algorithm is not the best of ever, it provides local optima of high quality for instances with many hundreds cities, within rather short times of execution. These ideas are currently being transferred to the quadratic assignment problem where we have exhibited different topologies ranging from landscapes with one massif central, many massif central and rugged. On the Job-Shop-Scheduling problem, we have found that the landscape is very rugged for the traditionnal operators. Rugged landscapes are typically landscapes that are hard to optimize because iterated local search algorithms are trapped after just a few steps on a local optimum. In particular, the landscape of the JSP contains (lots of) plateaus where all neighbors have the same fitness. Other means are necessary to alleviate this problem (see [8]).

We are currently working towards a better knowledge of the massif central as well as the vicinity in the case of the TSP. We are rather confident in the fact that this study will be useful in the case of the QAP. We are also studying $k \gg 3$-dimension TSP as well as non geometric instances. At the algorithmic level we are also considering path relinking techniques.

References

[1] Emile Aarts and Jan Karel Lenstra, editors. *Local search in combinatorial optimization.* John Wiley and Sons, 1997.

[2] V. Bachelet, Ph. Preux, and E-G. Talbi. Parallel hybrid meta-heuristics: Application to the quadratic assignment problem. In *Parallel Optimization Colloquium*, March 1996. available at www-lil.univ-littoral.fr/˜preux

[3] Vincent Bachelet. *Métaheuristiques parallèles et paysages : application au problème d'affectation quadratique.* PhD thesis, Université de Lille 1, LIFL, Villeneuve d'Ascq, (to appear, in french)

[4] Thomas Baeck, editor. *Proc. of the Seventh International Conference on Genetic Algorithms*, East Lansing, MI, USA, July 1997. Morgan Kauffman.

[5] Richard K. Belew and Lashon B. Booker, editors. *Proc. of the Fourth International Conference on Genetic Algorithms*, La Jolla, CA, USA, July 1991. Morgan Kauffman.

[6] Kenneth Boese. *Models for Iterative Global Optimization.* PhD thesis, UCLA, Los Angeles, CA, USA, 1996. available at http://vlsicad.cs.ucla.edu/˜boese/thesiscont.html.

[7] R.E. Burkard and S.E. Karish and F. Rendl. *QAPLib — a quadratic assignment problem problem library. Journal of Global Optimization*, **10**, 331–403, 1997. available at http://fmatbhp1.tu-graz.ac.at/%7Ekarisch/qaplib/qaplib.html

[8] D. Duvivier, Ph. Preux, C. Fonlupt, D. Robilliard, and E-G. Talbi. The fitness function and its impact on local search methods. In *Proc. Conf. IEEE System, Man, and Cybernetics*, pages 2478–2483, San Diego, USA, October 1998. IEEE Press.

[9] C. Fonlupt, D. Robilliard, Ph. Preux, and E-G. Talbi. Fitness landscape and performance of meta-heuristics. In *Meta-Heuristics — Advances and Trends in Local Search Paradigms for Optimization*, chapter 18, pages 255–266. Kluwer Academic Press, 1999.

[10] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the theory of NP-Completeness.* W.H. Freeman and Company, 1979. ISBN: 0-7167-1045-5.

[11] David S. Johnson and Lyle A. McGeoch. The traveling salesman problem: A case study. In *[1]*, pages 215–310. John Wiley and Sons, 1997.

[12] Terry Jones. *Evolutionary Algorithms, Fitness Landscapes and Search.* PhD thesis, University of New Mexico, Albuquerque, NM, USA, May 1995.

[13] M. Jünger, G. Reinelt, and G. Rinaldi. The traveling salesman problem. *Network Models*, 7:225–330, 1995.

[14] Stuart A. Kauffman. Adaptation on rugged fitness landscapes. In *[27]*, pages 527–618. 1988.

[15] Stuart A. Kauffman. Principles of adaptation in complex systems. In *[27]*, pages 619–712. 1988.

[16] S. Kirkpatrick and G. Toulouse. Configuration space analysis of travelling salesman problems. *J. Physique*, 46:1277–1292, August 1985.

[17] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnoy Kan, and D.B. Shmoys, editors. *The Traveling Salesman Problem — A Guided Tour of Combinatorial Optimization.* John Wiley and Sons, 1985.

[18] S. Lin and B. W. Kernighan. An effective heuristic algorithm for the traveling salesman problem. *Operation Research*, 21:498–516, 1973.

[19] Bernard Manderick, Mark de Weger, and Piet Spiessens. The genetic algorithm and the fitness landscape. In *[5]*, pages 143–150, 1991.

[20] O. Martin and S.W. Otto. Combining simulated annealing with local search heuristics. *Annals of operations research*, 63:57–75, 1996. Special issue Metaheuristics in Combinatorial Optimization, G. Laporte and I.H. Osman (eds).

[21] O. Martin, S.W. Otto, and E.W. Felten. Large-step markov chains for the TSP. *Operation Research Letter*, 1:219–224, 1992.

[22] Heinz Mühlenbein. Genetic algorithms. In *[1]*, pages 215–310. John Wiley and Sons, 1997.

[23] Yuichi Nagata and Shigenobu Kobayashi. Edge assembly crossover: A high-power genetic algorithm for the traveling salesman problem. In *[4]*, pages 450–457, 1997.

[24] J. Reinelt. TSPLIB: A traveling salesman problem library. *ORSA Journal on Computing*, 3:376–384, 1991. disponible à l'adresse http://www.iwr.uni-heidelberg.de/iwr/comopt/soft/TSPLIB95/TSPLIB.html.

[25] J.D. Schaffer, editor. *Proc. of the Third International Conference on Genetic Algorithms*, Bloomington, IN, USA, 1989. Morgan Kauffman.

[26] Peter F. Stadler and Wolfgang Schnabl. The landscape of the traveling salesman problem. *Physics Letter A*, 161:337–344, 1992.

[27] Daniel L. Stein, editor. *1988 Lectures in Complex Systems.* Santa Fe Institute Studies in the Sciences of Complexity. Addison-Wesley Publishing Company, 1989. SFI Studies in the Science of Complexity, Lectures Vol. I, ISBN: 0-201-52015-4.

[28] T.E. Vollmann and E.S. Buffa. The facility layout problem in perspective. *Management Science*, 12(10):450–468, 1966.

[29] E. Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics*, 63:325–336, 1990.

[30] Edward D. Weinberger. Local properties of kauffman's Nk model: a tunably rugged landscape. *Physical review A*, 44(10):6399–6413, November 1991.

[31] Darrell Whitley, Thimothy Starkweather, and D'Ann Fuquay. Scheduling problems and traveling salesman: The genetic edge recombination operator. In *[25]*, pages 133–140, 1989.