Integrating Exploratory Landscape Analysis into Metaheuristic Algorithms

Andreas Beham[1,2], Erik Pitzer[1], Stefan Wagner[1], and Michael Affenzeller[1,2]

[1]Heuristic and Evolutionary Algorithms Laboratory, School of Informatics, Communications and Media, University of Applied Sciences Upper Austria, Hagenberg Campus, Softwarepark 11, 4232 Hagenberg, Austria,
[2]Institute for Formal Models and Verification, Johannes Kepler University Linz, Altenberger Straße 69, 4040 Linz, Austria

July 23, 2018

Technical Report[1]

Abstract

The no free lunch (NFL) theorem puts a limit to the range of problems a certain metaheuristic algorithm can be applied to successfully. For many methods these limits are unknown a priori and have to be discovered by experimentation. With the use of fitness landscape analysis (FLA) it is possible to obtain characteristic data and understand why methods perform better than others. In past research this data has been gathered mostly by a separate set of exploration algorithms. In this work it is studied how FLA methods can be integrated into the metaheuristic algorithm. We present a new exploratory method for obtaining landscape features that is based on path relinking (PR) and show that this characteristic information can be obtained faster than with traditional sampling methods. Path relinking is used in several metaheuristic which creates the possibility of integrating these features and enhance algorithms to output landscape analysis in addition to good solutions.

1 Introduction

Approximation methods such as metaheuristics have proven to be suitable in providing good solutions in short time to a number of real-world problems, e.g. vehicle routing or scheduling [1]. Metaheuristics are often described as a rather general strategy that employs simple heuristics to identify good solutions

[1]The original source of the publication is available at `https://doi.org/10.1007/978-3-319-74718-7_57`. The final publication is available at link.springer.com.

in short time [12]. Due to their general description, they can be applied to a rather wide range of problems. Still, their general use is limited by the so called no free lunch (NFL) theorem which states that over all problems each algorithm results in the same average performance [13].

It is thus of research interest to identify which algorithms work better on which problems (and even problem instances). The algorithm selection problem (ASP) [9] describes this as the goal to select the best algorithm for each problem. Metaheuristics that are used by automated decision makers (ADM) in controlling real-world production or logistics processes cannot rely on human input. For every new problem instance the ADM often has little time to pick a method to solve it reasonably well. Some work has been published that describe the use of a database containing known problem instances and recorded performance of methods applied to them [11, 2]. This database can then aid in algorithm selection by relating the unknown problem instance to a known one. To calculate such a relation, fitness landscape analysis (FLA) is a promising method with which landscapes may be described by a set of features [8, 10].

1.1 Motivation: Automated Decision Making

In ADM scenarios a new problem instance is given to the decision maker that has to identify a good solution, usually given a fixed computational budget. The possibilities to handle these scenarios are (1) use a reasonable preconfigured default algorithm instance. This is a simple approach, but if the fitness landscapes vary to a large degree the default instance may not work very well and leave an overall mediocre impression. (2) employ meta-optimisation approaches such as F-race with a given portfolio of different optimisation algorithms [7]. This approach however may require more computational effort as it converges in the space of algorithm instances in addition to the actual solution space. Finally, (3) make use of algorithm selection (AS). This requires two phases

A "probing" the problem, i.e. exploring and characterising the landscape

B "solving" the problem, i.e. applying a suitable algorithm instance

Usually, the first phase also requires evaluating several samples from the solution space and computing the characteristics from this data. Thus, we may find good solutions only in phase B. If we denote the computational cost to complete phase A as R and that of the simple approach T we can state that approach (3) is suited when the specialised algorithm instance is able to identify solutions faster than $T - R$. This trade-off is shown graphically in Figure 1.

2 Exploratory Landscape Analysis

Obtaining these features is however not without cost. In order to obtain characteristic information of the landscape, exploratory analysis methods sample and evaluate it at least in some points. Previously described exploratory analysis methods such as random or adaptive walks describe a certain sampling strategy and associated features [8].

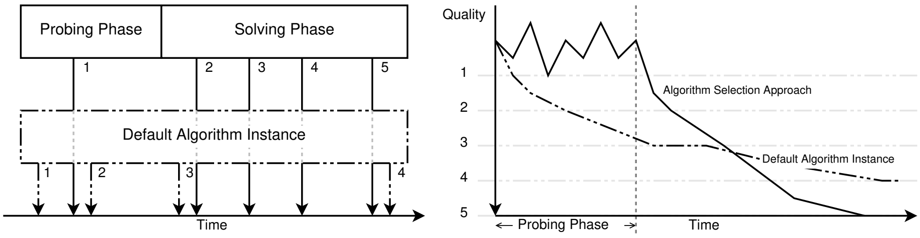

Figure 1: Schematic drawing of a scenario that favours algorithm selection compared to a simple strategy based on choosing a suitable default algorithm instance. The length of the probing phase is a disadvantage as the selected algorithm instance must be significantly faster than the default in order for this approach to be of use.

Random Walk

creates a trajectory of solutions by replacing the current solution with a random neighbour. It is assumed that neighbours have an identical and constant distance to each other, thus only the trajectory's fitness is used to obtain features. The so called quality trail represents the trajectory as a sequence of fitness values. Features that are computed from this trail are autocorrelation(1), correlation length, information content, partial information content, density basin information, information stability, diversity, regularity, total entropy, peak information content, peak density basin information. For more information on these the reader is referred to Pitzer et al. [8].

The advantage of random walks is that they are simple to describe and parallelise. One can compute all sampled solutions of the random walk in advance and obtain the fitness values to the solutions in parallel thereafter. The disadvantage is that visiting high fitness regions of the landscape is unlikely. Also the walk is unaware of concepts such as plateaus or local optima. However, landscape features such as funnels or distribution of local optima are perceived to be strongly connected to the success rate of algorithms [5].

n-Adaptive Walk

Adaptive walks attempt to sample solutions of better quality and thus descend into regions of the search space that are more likely metaheuristics will explore longer. An adaptive walk is achieved by replacing the current solution with the best from a neighbourhood sample of size n. The parameter n is crucial to the obtained characteristics. If the sample contains all neighbours, the adaptive walk is deterministic and essentially the same as a best improvement local search. It will most likely become trapped in a local optimum and thus has very limited capabilities for exploring the solution space. If n is set too low the sampled solutions will be similar to those obtained during random walks. Adaptive walks are slower to compute and may be parallelised only with respect to the sample size n. The same features can be obtained as in random walks, but they cannot be compared to one another [8].

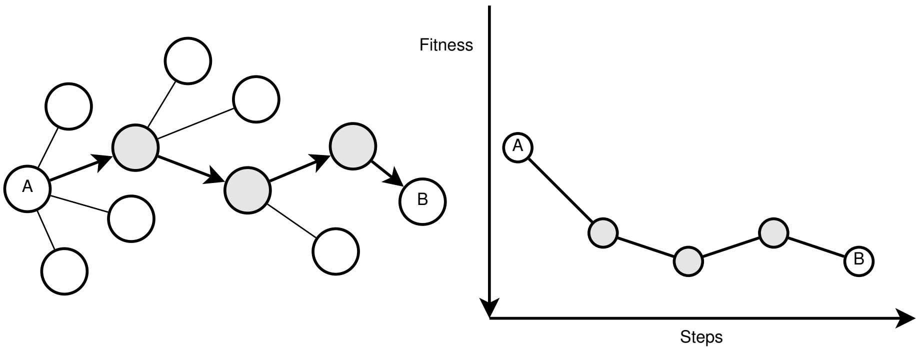

Figure 2: Path relinking between two solutions A and B.

3 Integrated Landscape Analysis

By integrating landscape analysis into metaheuristic algorithms we hope to embed the probing phase into the default algorithm instance run. So that when the default instance cannot identify better solution we can relate the problem instance to previously observed instances and make use of the recorded performance data of various other algorithm instances.

As we have shown earlier several walks exist for characterising landscapes. The characteristics however are mostly based on quality trails which may be difficult to reuse in population-based algorithms where such a trail does not exist. Also in trajectory-based algorithms such as local search the quality trail is only a very small result. For instance, the number of evaluated neighbours is often much larger than the iterations performed during local search.

3.1 Obtaining Characteristics from Path Relinking

Path relinking is a concept that could serve to generate useful data about the landscape. In path relinking we take two solutions that are different to each other and connect them through a best-improvement path of intermediate solutions.

Compared to local search the advantage of path relinking is that the neighbourhood becomes smaller the more similar the solution approaches the target. Another advantage is that plateaus in the solution space become visible easily. If non-deteriorating moves are present in the neighbourhood, they will be picked eventually when improving moves are no longer possible. The quality progress of the intermediate solutions thus becomes flat and highlights the presence of neutral areas. Detecting these regions is only possible with traditional walks if the sample size n is set to a rather high value. This however means that even more effort has to be spent for each iteration of the quality trail.

If we use randomly selected source and target solutions, we generally expect to see a "U-shaped" quality progress (assuming minimisation of the objective function). We see that these shapes may be representative for a certain fitness landscape. For instance, flat regions have already be mentioned, but also a rough objective function should result in a large changes to the progress and a curve that appears to be less smooth. In analysing these curves we have come up with three new characteristics These are based on calculating the differential of these curves, i.e. $\frac{\Delta y}{\Delta x}$. In this case Δy is the difference in fitness while Δx is the difference in solution similarity. The characteristics are:

- **Sharpness -** is the average value of the differential.
- **Bumpiness -** is close to represent the frequency of inflection points in curve analysis. Every time the differential changes sign an inflection point may be observed. If the differential changes sign often the landscape is considered bumpy or rough.
- **Flatness -** is close to represent the frequency of undulation points in curve analysis. This represents true flatness in the landscape.

If we assume a path p is an ordered set consisting of tuples of solution and associated fitness value $\{(p_1, r_1), (p_2, r_2), ..., (p_i, r_i), ..., (p_k, r_k)\}$ we calculate the differential in solution i as

$$\frac{r_{i+1} - r_{i-1}}{d(p_{i+1}, pi - 1)}$$

with $i > 1 \wedge i < k$ and where d is a distance function. By considering that r_i can also be the differential instead of the actual fitness value we may compute the differential of the differentials and thus something that is conceptually similar to the 2^{nd} derivative in curve analysis. This enables us to calculate bumpiness and flatness in that we observe the change of the differential over time.

4 Experiment Setup

We use 50 different instances of the quadratic assignment problem in this test. The problem instances were selected randomly based on various libraries, i.e. QAPLIB [3], Drezner [6], Microarray [4], and Taillard[2]. We chose to use only problem instances of the same dimension in order to avoid comparing large problem instances to small ones. Presumably this poses a more difficult situation for our test as we can rule out dimension as an a priori discriminatory factor. However, there are only very little instances available with exactly the same dimension. We thus chose a rather small dimension, i.e. 25, and reduced larger problem instances to this size. The reduction was performed by excluding randomly selected indices from the weights and distances matrix. We recreated these matrices without the respective rows and columns. Due to this unbiased reduction we hope to maintain the individuality of different problem instances. The full set of problem instances obtained after reduction has a size of 122 of which the 50 instances used in the test are randomly chosen. From the set of Taillard's instances we only choose the 20 instances of size 27 and reduce them to size 25. Otherwise Taillard's instances would highly dominate the benchmark set as they're basically stemming from the same generator and thus should have similar fitness landscapes.

The characteristics obtained by exploratory landscape analysis (ELA) vary from run to run. The hypothesis is that the characteristics of the landscape can be calculated with an increasing precision the more samples we obtain from

[2] `http://mistic.heig-vd.ch/taillard/problemes.dir/qap.dir/qap.html`

this landscape. It is also an interesting question how many such samples lead to which precision which has not yet received much attention in the literature.

We thus design a two-phased test. In the first "training" phase we draw several sets of samples each with an increasing size from the problem instances' landscape and calculate the characteristics. We then assume that these are the true characteristics of each landscape. In the second "testing" phase we draw different sets of samples, but of the same sizes, and then compute the characteristics anew. For each problem instance we then compute a ranking, based on the Euclidean distance between its test characteristics and the training characteristics. Finally, we record in the obtained ranking the position of the problem instance. Over all problem instances we then compute the average rank. A low average rank thus means that we can successfully relate a given problem instance to a set of previously known instances. The assumption is that we can obtain more precise characteristics given that the sample size increases and that we thus achieve lower ranks.

Table 1: Average rank of problem instances after comparing with characteristics obtained from a path relinking walk. Effort is given in number of paths. Each path required sampling and evaluating slightly less than 300 solutions.

		Testing Effort						
		1	5	10	20	50	100	200
Learning Effort	1	16.9	14.7	14.8	15.8	15.0	14.6	14.8
	5	14.3	10.6	7.7	7.4	7.3	7.3	7.3
	10	12.9	9.4	7.8	6.7	6.6	6.1	5.9
	20	12.6	9.3	6.6	7.4	5.3	5.7	5.4
	50	13.0	7.9	5.4	5.5	3.8	3.6	3.9
	100	13.6	7.7	5.5	5.0	3.8	3.6	2.8
	200	12.1	8.0	5.6	4.6	3.7	2.5	2.3

5 Results

The results given in Table 1 show that "directed walks" based on path relinking together with the three proposed features can achieve good results in characterising and recognising landscapes. With an increasing number of paths and effort spent in characterising the landscape the average rank decreases, i.e. the problem instance is found among the top ranked instances. This indicates that the paths from directed walks do indeed resemble unique properties of the landscapes. We also tried calculating the characteristics that have been described earlier (auto correlation, information content, etc.) given the trails of path relinking, but we could not achieve good results. Most likely the paths were too short for these characteristics, for example a problem instance of size 25 usually results in PR paths of length 22. And as we can see in the results of random and adaptive walks in Table 2 the trajectories need to be much longer. For random walks the length of the trajectory is equal to the effort, for adaptive walks the trajectory is the effort divided by the sample size n. We may also observe that for roughly the same amount of effort - one PR path takes on average 286 evaluated samples - directed walks based on path relinking perform better than both random and adaptive walks. However, we need to mention that we did not perform any feature selection and used all 11 features.

Table 2: Average rank of problem instances after comparing with characteristics obtained from a random and adaptive walks. Effort is given in evaluated solutions.

Learning Effort (x 10^3)	Testing Effort (x 10^3)													
	0.3	1.5	3.0	6.0	15.0	30.0	60.0	0.3	1.5	3.0	6.0	15.0	30.0	60.0
	Random Walk							10-Adaptive Walk						
0.3	10.6	10.9	10.8	13.2	13.9	13.5	14.3	14.5	20.3	20.5	21.9	22.4	22.6	22.6
1.5	12.6	6.7	5.9	7.1	7.3	8.8	9.8	16.2	9.9	11.3	15.4	17.0	18.2	18.7
3.0	13.8	8.9	7.5	6.4	7.6	8.4	9.2	16.2	10.0	8.8	10.9	12.7	14.9	15.7
6.0	11.2	6.8	5.4	5.9	5.4	6.9	7.3	16.9	11.6	8.4	5.9	6.8	10.8	11.7
15.0	13.2	7.8	6.2	4.7	4.1	4.8	5.7	17.4	14.2	11.4	9.0	7.4	6.3	9.6
30.0	12.3	8.7	7.4	6.7	4.7	4.2	4.5	17.6	14.1	12.2	9.8	5.8	5.2	6.6
60.0	12.0	7.9	6.9	5.7	4.8	4.6	3.5	17.8	13.9	12.5	11.0	7.5	6.2	5.5

6 Conclusions and Outlook

We have shown that we can obtain a good set of characteristics for recognising problem instances from path relinking. PR is a heuristic concept that is used in several algorithms which potentially enables obtaining these characteristics during the optimisation run. However, there are still quite a large number of samples necessary to achieve stable results. With 50 or more paths we observed a reasonable precision in the ranks of less than 4. This means that on average we can relate an instance to a very similar (in this case the same) instance within the top 4 similar problem instances.

Further work is necessary, especially with respect to "noisy" data. In metaheuristics that make use of PR, paths are often not only between randomly sampled solutions. This creates a more challenging scenario for the introduced features. In addition larger studies are necessary with respect to problem instances of higher dimensions. Additionally, as the proposed method is generic it should be compared on further problems.

Acknowledgements

The original source of the publication is available at `https://doi.org/10.1007/978-3-319-74718-7_57` .
The final publication is available at `link.springer.com`.

References

[1] Affenzeller, M., Winkler, S., Wagner, S., Beham, A.: Genetic Algorithms and Genetic Programming - Modern Concepts and Practical Applications. Numerical Insights, CRC Press (2009)

[2] Beham, A., Wagner, S., Affenzeller, M.: Optimization knowledge center: A decision support system for heuristic optimization. In: Companion Publication of the 2016 Genetic and Evolutionary Computation Conference, GECCO'16 Companion. pp. 1331–1338. ACM (2016)

[3] Burkard, R.E., Karisch, S.E., Rendl, F.: QAPLIB – A quadratic assignment problem library. Journal of Global Optimization 10(4), 391–403 (June 1997), http://www.opt.math.tu-graz.ac.at/qaplib/

[4] de Carvalho Jr., S.A., Rahmann, S.: Microarray layout as quadratic assignment problem. In: Proceedings of the German Conference on Bioinformatics (GCB), volume P-83 of Lecture Notes in Informatics (2006)

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

[6] Drezner, Z., Hahn, P.M., Taillard, E.D.: Recent advances for the quadratic assignment problem with special emphasis on instances that are difficult to solve for meta-heuristic methods. Annals of Operations Research 139, 65–94 (2005), https://han.ubl.jku.at/han/springerlinkdb/www.springerlink.com/content/xv4417r60p513774/fulltext.pdf

[7] López-Ibáñez, M., Dubois-Lacoste, J., Stützle, T., Birattari, M.: The irace package, iterated race for automatic algorithm configuration. Tech. Rep. TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium (2011), http://iridia.ulb.ac.be/IridiaTrSeries/IridiaTr2011-004.pdf

[8] Pitzer, E., Affenzeller, M.: Recent Advances in Intelligent Engineering Systems, chap. A Comprehensive Survey on Fitness Landscape Analysis, pp. 161–191. Springer Berlin Heidelberg, Berlin, Heidelberg (2012), http://dx.doi.org/10.1007/978-3-642-23229-9_8

[9] Rice, J.R.: The algorithm selection problem. Advances in Computers 15, 65–118 (1976), http://www.sciencedirect.com/science/article/pii/S0065245808605203

[10] Richter, H., Engelbrecht, A.: Recent advances in the theory and application of fitness landscapes. Springer (2014)

[11] Scheibenpflug, A., Wagner, S., Pitzer, E., Affenzeller, M.: Optimization knowledge base: An open database for algorithm and problem characteristics and optimization results. In: Companion Publication of the 2012 Genetic and Evolutionary Computation Conference, GECCO'12 Companion. pp. 141–148. ACM (2012)

[12] Talbi, E.G.: Metaheuristics: From Design to Implementation. John Wiley & Sons (2009)

[13] Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1(1), 67–82 (1997)