Visualization for Analyzing Trajectory-Based Metaheuristic Search Algorithms

Steven HALIM and **Roland H.C. YAP**[1] and **Hoong Chuin LAU**[2]

1 INTRODUCTION

Most combinatorial (optimization) problems are computationally intractable. We often have to be satisfied with good solutions and typically metaheuristic algorithms are used such as various forms of local search, tabu search, etc. Given the heuristic nature of such search algorithms, there are two important considerations in designing a metaheuristic algorithm:

- choice of metaheuristics to employ, which may include problem specific tweaks;
- selecting the appropriate parameters to drive the heuristics.

We call this problem of designing the appropriate metaheuristic problem for a combinatorial (optimization) problem, the *metaheuristic tuning problem* [1, 6]. Anecdotal evidence suggests that that tuning takes a major effort, i.e. [1] cites anecdotal evidence that 90% of the time is spent on the tuning problem.

Although it can be easy to come up with a variety of metaheuristics, tuning the metaheuristic implementation is not straightforward. Firstly, the metaheuristics may not be well understood. It might also be applied to problems which may not have been studied. Thus, it may not be clear how to perform tuning. Secondly, the space in which the tuning can operate on is very rich — there are many ways of combining different kinds of metaheuristics each with their own choice of strategies and variations. Furthermore, they each have their own parameters. In this paper, we take a broad view of the metaheuristic tuning problem and understand it to also encompass algorithm design and debugging.

Traditionally, the approach for to the tuning problem is either manual experimentation or more automated approaches such as finding the best parameter values [1], best configuration [3], or have self-tuning algorithms [2]. In this paper, we take a different approach which takes a human/programmer perspective — how to aid the human in dealing with the tuning problem. Like human-guided search [5] we believe that a cooperative paradigm whether the human is in the loop can be productive, the difference is that human-guided search is concerned with using the human to produce better solutions while we want to use the humans to produce better metaheuristic algorithms.

Ultimately, we would like a man-machine cooperation which can help the human debug, analyze and improve a metaheuristic algorithm for particular problems. Some of the questions which we would like to help answer are:

- Does the search behavior match how we think the algorithm should behave?
- Are there signs of cycling behavior?
- How does the metaheuristic algorithm make progress?
- How effective the metaheuristic in conducting intensification and / or diversification?
- How wide is the search coverage?
- How far is the (greedy) initial solution to the best found solution?
- Does the search quickly identify the region where best found solution reside or wander around in other regions?
- How do the trajectories of two different metaheuristics compare?
- What is the effect of modifying certain parameters, components or strategies with respect to the search behavior?

In this paper, we focus on metaheuristic algorithms which are search trajectory based, such as iterated local search, simulated annealing, tabu search, etc. We believe that a good approach to get man-machine cooperation is with an interactive visualisation of the search trajectory. One way of understanding how a program works is with a debugger. We have built the analog of a debugger, the visualizer VIZ, for understanding search trajectories of metaheuristic algorithms by providing visualization and animation (forwards and backwards) in time. VIZ has both problem independent visualization which allows it to be used on a broad range of metaheuristic algorithms in a generic way and can also make use of problem specific visualizations. Although VIZ is still in prototype stage, we believe that it is the first serious attempt at an interactive tool with emphasis on the human computer interaction aspects to help humans understand the dynamic behavior of metaheuristic algorithms and guide the tuning process.

2 SEARCH TRAJECTORY VISUALIZATION

Visualizing the search trajectory, i.e. local movement of the current solution along the search space is difficult because the problem is usually in very high dimensions and the search space is also extremely large. We are only aware of (very) few proposals for search trajectory visualization.

N-to-2 space mapping [4] gives a mapping from a higher dimensional problem space to 2-D for visualizing, e.g. coverage of search space. However, the proposed visualization is crowded and static.

In earlier work, V-MDF [6], we proposed a visualization called the *distance radar*. A current set of elite solutions is chosen, called *anchor points*. The distance radar displays two graphs: (i) the distance between the current solution in the search trajectory to the set of anchor points; and (ii) the fitness with respect to the anchor points. While V-MDF can help answer some of the questions about how the search trajectory is behaving, the visualization is not very intuitive. For example, one drawback is that the graphs can change simply because the elite set changes with time. The visualisation is

[1] National University of Singapore, {stevenha,ryap}@comp.nus.edu.sg
[2] Singapore Management University, hclau@smu.edu.sg

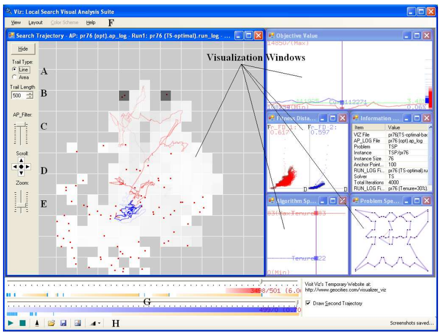

Figure 1. Screen shot of VIZ with multiple visualizations

less effective here because essentially one dimensional information, the distance or objective value is displayed for every anchor point.

With VIZ, we want to ensure that the visualization can be intuitive and exploit the fact that humans are good at recognizing visual patterns, in particular, not just an image but also how how things change which exploits movement and temporal features. We make use of a 2-D visualisation where instead of trying to map from the points in the search trajectory from a high dimension to 2-D, we consider abstract points which are differentiated from each other using a distance metric. We do not have the space to discuss the visualization in detail — an example of a possible metric is the Hamming distance. We can then then layout these points in 2-D as shown in Fig. 1 and 2. The advantage is that we now have a problem independent visualization which can be used with a suitably defined metric to demonstrate search trajectories.

3 The VIZ system

Fig. 1 shows VIZ — the GUI, the independent visualizations are the large search trajectory visualization and search trajectory objective value graph, and other problem specific visualizations (i.e. current solution as a TSP tour). VIZ functions in interactive fashion as a kind of video player to play back an animation of the search trajectory (forwards/backwards) drawn as a trail. The trail fades with time so that the animation does not clutter up the screen. Colors are used to compare two metaheuristic algorithms. Anchor points[3] are landmarks to indicate the progress of the search trajectory in the abstract search space. The problem specific visualizations are used to complement the problem independent ones. The animation of the geometric pattern of the trail and its relation to the anchor points and problem specific visualizations can be used to answer the questions in Sec. 1.

Space doesn't permit more details, rather we use the following example which demonstrates how one can visualize the differences between two variants of Iterated Local Search on TSPs [8]. In TSPs, it is conjectured that a heuristic algorithm should exploit the "Big Valley" property, a region in the TSP search space where most local optima (including the global optima) lie [7].

In this example, we want to know whether our algorithms make use of this property. we created two variants of the ILS algorithm which are run on the same TSP instance: ILS_A and ILS_B. The visualization of the search trajectories from ILS_A and ILS_B is shown in Fig. 2. We can see that at a glance, one can check the existence of the search intensification indicative of a 'Big Valley' by looking for

[3] These are different from V-MDF, as we do not want anchor points to move. The size of the anchor points indicate the fitness.

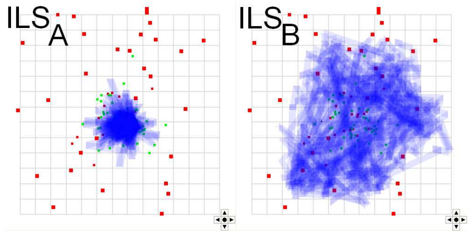

Figure 2. Search trajectory of ILS_A vs ILS_B on the same TSP instance

a search trajectory clustered together with improving moves. Fig. 2 shows that the search trajectory for ILS_A is concentrated in the middle of the screen (indicative of the 'Big Valley' area). ILS_B, on the other hand, uses too much diversification. The length of the edges in the trail of ILS_B indicates that it jumps 'quite far'. It is also seldom exploring the regions near the best anchor points.

This gives a possible explanation of the algorithm behavior and suggests directions for tuning out algorithms. If our solution behaves like ILS_A, we know that we are on the right track, perhaps only few minor adjustments are needed. In the other hand, if it behaves like ILS_B, we may want to modify our ILS algorithm such that it is more focused.

4 CONCLUSION

We have presented a new approach for visualizing search trajectory and introduced the visualizer tool VIZ. This is not intended to replace the existing analysis tools, but rather to augment the existing tools to help the algorithm designer better understand the behavior of a trajectory based metaheuristic search algorithm and to debug and tune the the algorithm. A prototype of VIZ which is under continuous development is at: `www.comp.nus.edu.sg/~stevenha/viz`

ACKNOWLEDGEMENTS

This paper was written while Roland Yap was visiting the Swedish Institute of Computer Science and their support and hospitality are gratefully acknowledged.

REFERENCES

[1] B. Adenso-Diaz and M. Laguna, 'Fine-tuning of Algorithms Using Fractional Experimental Designs and Local Search', *Operations Research*, (2005).

[2] R. Battiti and G. Tecchiolli, 'The Reactive Tabu Search', *ORSA Journal of Computing*, **6(2)**, 126–140, (1994).

[3] M. Birattari, *The Problem of Tuning Metaheuristics as seen from a machine learning perspective*, Ph.D. dissertation, University Libre de Bruxelles, Belgium, 2004.

[4] M. Kadluczka, P.C. Nelson, and T.M. Tirpak, 'N-to-2 Space Mapping for Visualization of Search Algorithm Performance', in *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, pp. 508–513, (2004).

[5] G.W. Klau, N. Lesh, J. Marks, and M. Mitzenmacher, 'Human-Guided Tabu Search', in *Proceedings of the 18th National Conference on Artificial Intelligence*, pp. 41–47, (2002).

[6] H.C. Lau, W.C. Wan, and S. Halim, 'Tuning Tabu Search Strategies via Visual Diagnosis', in *Proceedings of the 6th Metaheuristics International Conference*, (2005).

[7] P. Merz, *Memetic Algorithms for Combinatorial Optimization Problems: Fitness Landscapes and Effective Search Strategies*, Ph.D. dissertation, University of Siegen, Germany, 2000.

[8] T. Stuetzle and H. Hoos, 'Analyzing the Run-Time Behavior of Iterated Local Search for the TSP', in *Proceedings of the 3rd Metaheuristics International Conference*, (1999).