Contents

Chapter 5

Integrated White+Black Box Approach

Two are better than one, because they have a good return for their work.
If one falls down, his friend can help him up.
But pity the man who falls and has no one to help him up.[1]
— **Ecclesiastes 4:9-10 (Holy Bible) [15]**

In this chapter, we discuss the importance of maximizing human+computer collaboration in addressing SLS DESIGN AND TUNING PROBLEM. *We combine both the white-box approach (heavily uses human strengths) and the black-box approach (heavily uses computer strengths). This chapter has been published in [6].*

5.1 The Components

5.1.1 White-Box: Fitness Landscape Search Trajectory Visualization

In Chapter 4, we have shown that understanding the characteristics of the fitness landscape empowers the algorithm designer to tailor the SLS implementation so that the search performs a better trajectory on the fitness landscape. However, matching an SLS algorithm to the fitness landscape(s) of an instance or class of instances of a COP is not easy:

1. Different problem instances of the same COP (characteristics, instance sizes, etc) may have quite different fitness landscapes [7].
2. The SLS behavior depends on the fitness landscape [7, 10, 12]. SLS with the same configuration may behave differently on different fitness landscapes.
3. The selected configuration (parameters [3, 1], heuristics components [4], and search strategies [5]), the implementation details, and any unexpected programming bugs, determine the *actual* SLS behavior. This behavior may be wrong, e.g. doing diversification when we expect the SLS to do intensification on the fitness landscape (failure modes [14]).
4. Stochastic elements mean the SLS can take different search trajectories in replicated runs.

We have seen the limitation of the existing white-box approaches in Chapter 4.4. Then, in Chapter 4.5, we show how the abstract fitness landscape search trajectory (FLST) visualization can enhance the understanding of the SLS behavior on the COP fitness landscape.

Our integrated white+black box approach uses this FLST visualization to first *understand* the characteristics of the fitness landscape of the COP (e.g. the fitness landscape is rugged). This helps the algorithm designer in *predicting* which search strategies will likely work well on the fitness landscape of the COP instance. The prediction can then be *verified* via FLST visualization to see whether the SLS encounters any 'problem(s)' (e.g. stuck in a local optima). The algorithm designer uses these observations to produce insights (which may be 'outside the box'), to *make informed changes* (e.g. add strong diversification strategies) and also to *narrow down the possible configuration space* (e.g. avoid SLS configurations that make the SLS harder to escape from local

[1] Our thesis is that cooperation man and machine is fruitful for addressing SLS DESIGN AND TUNING PROBLEM.

optima such as lowering the tabu tenure in Tabu Search). It is hard to arrive at these decisions without proper white-box approaches.

The white-box component of Integrated White+Black Box Approach leverages on the strengths of humans to analyze and learn from the FLST visualization. Although this is subjective, we show later in Chapter 7 that this process is intuitive and fruitful insights can be gained.

5.1.2 Black-Box: Full Factorial Design

Ideally, given an initial SLS configuration space, black-box tuning algorithms can be used to systematically find the most suitable configuration in the given configuration space to attack the COP at hand. However in practice, the size of the SLS configuration space size may be huge. As such, the algorithm designer must give a 'sufficiently narrow' configuration space for the black-box tuning algorithm to work with since tuning time would otherwise take too long. Furthermore, if the best configuration happens to be 'outside the box' (the initial configuration space), then it cannot be found by fine-tuning alone.

Our integrated white+black box approach combines the strengths of both approaches to complement their weaknesses. After gaining insights into the fitness landscape and the SLS behaviors on the fitness landscape via FLST visualization, the algorithm designers can use the insights to tweak the design of SLS, either by using *known* or *new* heuristic tweaks (e.g. adding a strong diversification strategy). This narrows down the configuration space substantially. The new algorithm can then be more easily fine-tuned (e.g. precisely how much diversification).

While a white-box approach may be usable for designing good SLS algorithm, it is still tedious for humans to explore the narrowed configuration space manually. Automatic black-box tuning algorithms are best for this situation. We have chosen to implement a *full factorial design* [11] in Viz system since we can obtain smaller configuration spaces through the white-box visualization process. However, other black-box tuning tools such as ParamILS [8], F-Race [3], CALIBRA [1], or iMDF [9] can be used in this phase.

5.2 The Integrated Approach

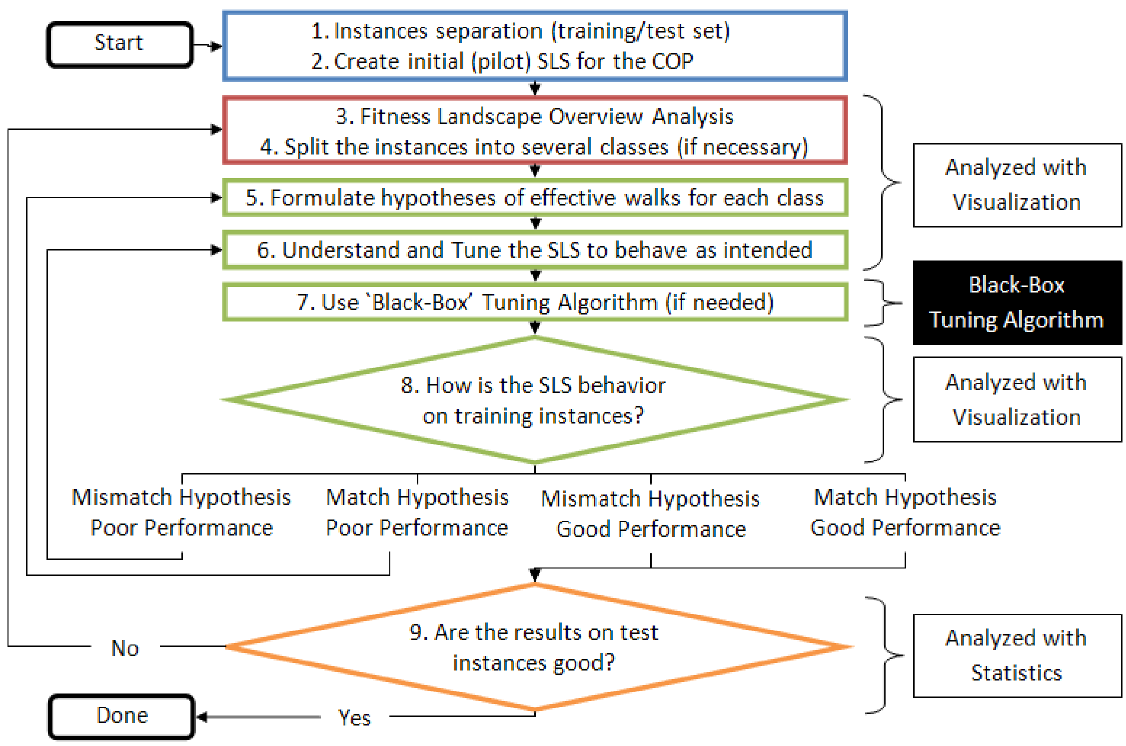

Figure 5.1: The Integrated White+Black Box Approach Methodology (see details in the box below)

The methodology of the integrated white+black box approach is summarized in Figure 5.1. While the general approach is not new (typical for white-box approaches, e.g. [2]), what is novel here is how FLST visualization (see Chapter 4) has been integrated in the white-box steps 3–6, 8, and black-box tuning algorithm in the black-box step 7. As we will show, the use of visualizations in Viz (see Chapter 6 for details about Viz) makes it much easier to analyze the fitness landscapes and SLS behaviors. Viz also has integrated support for black-box tuning in step 7 and some other handy automation for running SLS experiments and computing statistical analysis in step 9. The roles taken by human and computer in the integrated approach is maximized (see Table 5.1).

Approach	Human Role	Computer Role
Black-Box	Minimized	Maximized
White-Box	Maximized	Minimized
Integrated	Maximized	Maximized

Table 5.1: Human and Computer Roles in Different Approaches, also see Appendix C.

The more elaborated description of Figure 5.1 is as follows:
Note: this is not a strict guideline.

Step 1-2 : Algorithm designer implements an SLS algorithm *that works* for the given COP (pilot implementation).

Step 3-5 : With the pilot implementation at hand, the algorithm designer runs some pilot runs to understand the COP Fitness Landscape(s) by answering questions posed in Chapter 4.2. He splits the instances into classes if significant differences are found. He then formulate *hypothesis* of effective walks for each class of instances.

Step 6 : The algorithm designer designs an experiment to answer questions posed in Chapter 4.3. He uses FLST visualization to observe various SLS behavior on some COP training instances. The FLST visualization is strongly required here as these kind of information is hard to be understood without proper visualization. The obtained insights from the understanding of how the algorithm *actually* behaves greatly increase the chance for the algorithm designer to design better SLS algorithm or to invent new strategies/ideas.

Step 7 : The white-box step in step 6 may also narrow down the configuration space substantially. The algorithm designer can then pass the focused configuration space to a black-box tuning algorithm for more performance. Human judgment is usually not optimal here, we can ask computer to be more objective in this case.

step 8 : Analyze the results w.r.t to hypothesis of effective walks and the actual SLS performance. There are several possibilities:

- **If match hypothesis and good performance** : The SLS algorithm is currently ok for the training instances. Do verification on test instances!
- **If mismatch hypothesis but good performance** : Perhaps this is a discovery? Verify if this can be reproducible or just luck?
- **If match hypothesis but bad performance** : Perhaps the hypothesis is wrong.
- **If mismatch hypothesis and bad performance** : Adjust the SLS again.

Step 9 : Verify the results on test instances. If satisfactory SLS algorithm performance is achieved on test instances, stop. Otherwise, go back to step 3 as perhaps this is a case of over-fitting to training instances.

5.3 Simple Illustration

We apply this integrated approach in details in Chapter 7 Here, we *briefly* illustrate an example of using the integrated approach on the Traveling Salesman Problem (TSP, see Appendix A). The chosen SLS is Iterated Local Search (TSP-ILS, see Appendix B). TSP-ILS performs a 4-Opt double bridge perturbation, then its move operator swaps several pair of tour edges to reach a new 2-Opt TSP local optima. If it is better, TSP-ILS will move to the new local optima [13].

In Figure 5.2, we see that good quality (blue circle) and medium quality (green triangle) APs form one big cluster in the middle of the visualization (shown by the solid black arrows) and are

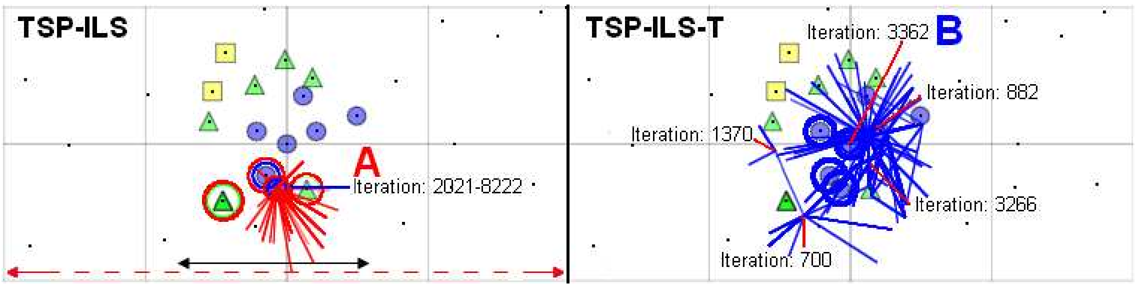

Figure 5.2: Visualization of TSP Fitness Landscape and ILS Behavior. See text for details.

close to each other when compared with the diameter of the fitness landscape (*partially* shown by the dashed red arrows). This shows a well known phenomenon called 'Big Valley'. We observe that outside this Big Valley region, we mostly see very bad (tiny black dot) APs. For such fitness landscape, it is suggested that the SLS should simply concentrate on the Big Valley region rather than wandering too far from it [7, 10, 13].

TSP-ILS already uses this strategy. However, FLST visualization shows that sometimes it is stuck in a local optima and unable to escape. Animation reveals that TSP-ILS is stuck in a place shown in Figure 5.2, label 'A'. This phenomenon is also observable with RTD analysis by [13]. In [13], the authors suggested to use a stronger diversification than 4-Opt: 'FDD-diversification' after a cut-off time has elapsed without any improvement. Without going into details, the improved behavior is observable in Figure 5.2, label 'B' where the tweaked TSP-ILS-T is now able to escape from several local optima attractors and progresses closer towards the center of the screen (the best-known solution). With white-box analysis, one can derive reasonable variants of FDD-diversification. The range of the cut-off time can be predicted using white-box approaches like the RTD or visualization analysis above but the exact value is best determined with black-box tuning.

5.4 Summary

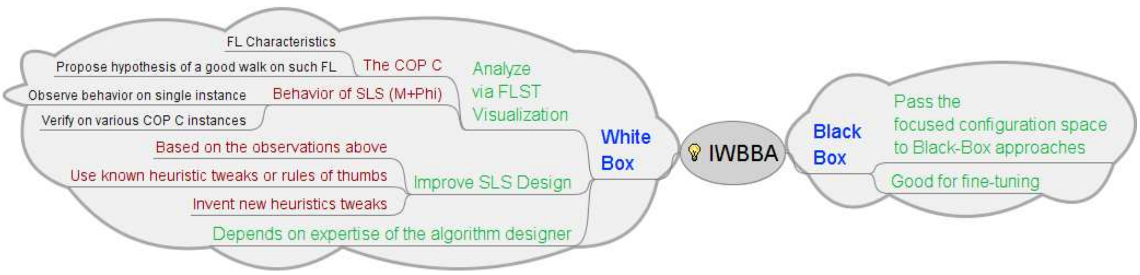

Figure 5.3: Chapter Overview

1. Understanding the behavior of SLS algorithm on different COP fitness landscape helps the algorithm designer in designing appropriate search strategies to tackle those COP instances.

2. While human is good in designing good SLS algorithms, picking the most optimal configuration in the focused configuration space (fine tuning) is still best done by computer.

3. We claim that INTEGRATED WHITE+BLACK BOX APPROACH is good to attack 'new' COPs (which are usually variants of the known COP).

5.5 Further Discussions

In the next chapter, Chapter 6, we present VIZ: a tool that summarizes all our discussion in Chapter 4 and in this Chapter 5 plus many other extra stuffs. The results of using INTEGRATED WHITE+BLACK BOX APPROACH on various COPs are presented in Chapter 7.

Bibliography

[1] Belarmino Adenso-Diaz and Manuel Laguna. Fine-tuning of Algorithms Using Fractional Experimental Designs and Local Search. *Operations Research*, 54(1):99–114, 2006.

[2] Thomas Bartz-Beielstein. *Experimental Research in Evolutionary Computation: The New Experimentalism*. Springer, 2006.

[3] Mauro Birattari. *The Problem of Tuning Metaheuristics as seen from a machine learning perspective*. PhD thesis, University Libre de Bruxelles, 2004.

[4] Irène Charon and Olivier Hudry. Mixing Different Components of Metaheuristics. In *Meta-Heuristics: Theory and Applications*, pages 589–603. Kluwer Academic Publishers, 1996.

[5] Steven Halim and Hoong Chuin Lau. Tuning Tabu Search Strategies via Visual Diagnosis. In *Meta-Heuristics: Progress in Complex Systems Optimization*, pages 365–388. Kluwer Academic Publishers, 2007.

[6] Steven Halim, Roland Hock Chuan Yap, and Hoong Chuin Lau. An Integrated White+Black Box Approach for Designing and Tuning Stochastic Local Search. In *Principles and Practice of Constraint Programming*, pages 332–347, 2007.

[7] Holger H. Hoos and Thomas Stützle. *Stochastic Local Search: Foundations and Applications*. Morgan Kaufmann, 2005.

[8] Frank Hutter, Holger H. Hoos, and Thomas Stützle. Automatic Algorithm Configuration based on Local Search. In *National Conference on Artificial Intelligence*, 2007.

[9] Hoong Chuin Lau and Fei Xiao. Toward an Intelligent Metaheuristics Framework. In *7th Metaheuristics International Conference*, 2007.

[10] Peter Merz. *Memetic Algorithms for Combinatorial Optimization: Fitness Landscapes & Effective Search Strategies*. PhD thesis, University of Siegen, Germany, 2000.

[11] NIST. e-Handbook of Statistical Methods.
http://www.itl.nist.gov/div898/handbook.

[12] Johannes J. Schneider and Scott Kirkpatrick. *Stochastic Optimization*. Springer, 2006.

[13] Thomas Stützle and Holger H. Hoos. Analyzing the Run-Time Behavior of Iterated Local Search for the TSP. In *3rd Metaheuristics International Conference*, pages 449–453, 1999.

[14] Jeal-Paul Watson. On Metaheuristics "Failure Modes". In *6th Metaheuristics International Conference*, pages 910–915, 2005.

[15] Zondervan. *New International Version (NIV) Holy Bible*. Zondervan, 1978.

Index