Contents

Chapter 1

Introduction

A journey of a thousand miles must begin with a single step.[1]
— **Lao Tzu**

Here, we present the thesis as succinctly as possible. For precise details, the readers are encouraged to probe the individual chapters. We will re-write this chapter again at the end of the author's PhD candidature because actually this chapter should be written 'last' after all else are written!

1.1 The SLS Design and Tuning Problem

$\mathcal{NP}$-hard Combinatorial (Optimization) Problems (Chapter 2.1)

In our life, all of us face all sorts of *computational problems*. Some can be solved 'easily' by ourself, e.g. $11 * 7 = 77$. For some other problems, we may prefer to have the help from computers, e.g. computing $f(x, y) = x^7 + xy^2 + \sin(x.\pi)/20.7 + 5$ for $x = 0.5$ and $y = 7^2$.

It is a notorious fact that computers have helped us to solve many important computational problems that are too overwhelming to be solved by hand. However, contrary to what commonly believed by *laymen*, there are still *many* other computational problems that are not yet possible to be exactly solved in *reasonable time* even by using today's fastest parallel super computer.

The difficult computational problems that we are dealing with in this thesis belong to the class of $\mathcal{NP}$-hard Combinatorial (Optimization) Problems (COPs). In this type of problem, we are interested to find the best out of (extremely) many possible *combinatorial* solutions of a given COP, where the 'best' is defined by the user's objective function.

COPs are found in *many* practical applications and thus solving these COPs efficiently is of paramount importance. The most naïve algorithm to solve COPs is to enumerate all possible combinations for the solution of the COP and simply pick the best one. However, due to the combinatorial nature of these COPs, the number of possible solutions of a given COP instance grows exponentially with the instance size, thus virtually renders any attempts for solving these COPs using naïve algorithms useless, especially for large instances.

Attacking $\mathcal{NP}$-Hard COPs (Chapter 2.2 - 2.3)

Computer scientists have devised better ways than naïve enumeration algorithm to find guaranteed optimum solution(s) for these COPs. Example of such *exact* algorithms are 'Branch & Bound' or 'Branch & Cut'. Notwithstanding their capabilities, these algorithms still have *very* high computational complexity.

Mainly due to this complexity issue, people usually resort to *non-exact* approaches for attacking COPs: sacrifice the guarantee of finding optimal solution in order to find 'good enough' solutions in *reasonable* amount of computation time.

[1] As with other thesis, this thesis is also the author's 'journey of thousand miles', possibly the longest scientific document that the author have ever written. This chapter marks its humble beginning.

[2] To satisfy the reader's curiosity, the answer is $f(0.5, 7) = 29.556121$

In the last few decades, a form of such non-exact algorithms called Stochastic Local Search (SLS), a.k.a Metaheuristics, emerges as potential candidate for attacking COPs. Basically, SLS works by *iteratively* searching (or constructing) potential combinatorial solutions *locally*, with no guarantee of optimality but usually can produce reasonably good solutions in short running time.

The Higher-Level SLS Design and Tuning Problem (Chapter 3.1 - 3.3)

Creating a simple SLS that works for a given COP is easy. All one needs to do is to instantiate certain SLS components, set some parameters with (usually) default values, and run the algorithm on a set of COP instances. This process has been further simplified by using software frameworks or class libraries[3].

However, while creating a simple working SLS *that works* for a given COP is easy, to make it performs *well* is *difficult* — a totally different story. An implementation of an SLS is considered successful if it can obtain *acceptable* quality solutions over a given set of COP instances, quite *robust*, and do so in *a reasonable amount of time*. Only in this sense, then this SLS algorithm is said to edge out exact algorithms for a given COP in practical cases.

Here is the hard part: The performance of SLS algorithm for solving a given $\mathcal{NP}$-hard COP depends on many usually correlated factors. Different COPs or even instances of the same COP often requires customized and holistic configuration of the SLS algorithm (set of search parameters, components and search strategies) so that the SLS algorithm performs well, otherwise the SLS performance is 'average' or even 'poor'. In some cases, we cannot just use the current off-the-shelf SLS algorithm as its initial performance is so poor and must be redesigned to better suit the COP at hand. This is what we termed as the SLS Design and Tuning Problem.

It is not easy to address the SLS Design and Tuning Problem as many things about the SLS or the COPs that are not well understood yet. Some algorithm designers, influenced by their past knowledge and experiences, resort to trial-and-error approach through some random experiments – which is quite a bad idea, or through well designed experiments – which is better but still laborious. Others created 'parameterless' SLS, in which the algorithms will self adjust itself upon detecting some 'events' during the search. While this self-correcting (a.k.a reactive) strategy is interesting, it is not trivial to come up with a good self-correcting strategy for any given situation. From the industry standpoint, many of these traditional approaches are unproductive especially against a backdrop of tight development schedule.

The SLS Design and Tuning Problem is encountered by the researchers and practitioners *every time* they develop an SLS for a given COP. Typically, tackling the SLS Design and Tuning Problem constitutes the bulk of the SLS development time. Therefore, if we have a better way to address this important issue, we can save a lot of development time required to build a sophisticated SLS implementation given any COP.

Various Clever Ways to Address SLS Design and Tuning Problem (Chapter 3.4 - 3.8)

To date, there are several researchers working on this SLS Design and Tuning Problem issue. Out of several approaches, we classify the more general ones into two parts: black-box or white-box approaches.

Black-box approaches aim to create tuning algorithms to automatically configure SLS algorithms by *semi*-exhaustively explore the configuration space and return the best found configuration within the given configuration time limit. This is an obvious improvement over manual trial-and-error as this time computer is the one doing the tedious trial-and-error for us, often by employing some experiment design techniques such that only promising potentially configurations are tried more thoroughly.

In the other hand, white-box approaches are techniques, methods, or tools that are created to assist the algorithm designer in designing better SLS algorithm by opening up the 'box', allowing the algorithm designers to inspect the inner working of the SLS algorithm. This can assist the algorithm designers in coming up with good search strategies or in choosing appropriate SLS components/parameter values.

Previously, no single approach is clearly superior to address *all* types/aspects of SLS Design and Tuning Problem. Black-box approaches are simple to apply, but will not help if the best configuration happens to be 'outside the box' of the initial configuration space. White-box

[3]As a note, we have build one such SLS software framework in the past (MDF) [2, 1]

approaches can give the algorithm designer insights into the search process, but are less effective for fine-tuning.

1.2 Our Proposed Approaches and Tools

Visualization to explain SLS behavior on COP Fitness Landscape (Chapter 4 & 6)

To tackle this SLS Design and Tuning Problem issue, we propose a collaboration between human[4] and computer. This collaboration is possible because human (e.g. intelligence, visual perception abilities, etc) and computer (e.g. speed, consistency, etc) complements each other, with information visualization as the bridging interface between them.

The **information visualization** interface is in form of a visualization of a COP fitness landscape and the SLS trajectory on it. It is known in the literature that the fitness landscape structure of a COP instance affect the behavior of SLS algorithm that is running over it. To design an effective SLS for a particular COP, a preliminary fitness landscape analysis of various instances of the COP is almost compulsory. However, even the fitness landscape of moderate-size COP instance is already too big to be exhaustively explored (exponential size). Various standard analytical metrics are inadequate to analyze the search trajectory details of SLS runs. A better way is needed.

We have invented a visualization technique called Fitness Landscape and Search Trajectory (FSLT) visualization, explained in depth in Chapter 4. We have also build a concrete off-line visualization tool Viz which incorporates our proposed FLST visualization techniques above, coupled with existing statistical and information visualization techniques. The details of this tool is presented in Chapter 6.

The Integrated White+Black Box Approach (Chapter 5)

However, as said before, white-box approach in form of visualization alone is not suitable for fine tuning the SLS parameters. Thus, we develop the Integrated White+Black Box Approach, where we first start with an SLS that works for a given COP and seek to understand the SLS behavior on the fitness landscape of the various instances of the given COP via a well designed visualization tool (white-box). Then, we use the knowledge to roughly tune the SLS (add potential search strategies, choose appropriate SLS components, or set good parameter values range). The SLS is then fine-tuned using black-box factorial design technique. Integrated White+Black Box Approach utilizes the strengths of both white-box and black-box approaches to produce better results than either alone.

The Results (Chapter 7)

We applied our approaches and tools to design and tune various SLS implementations for several COPs. The results are promising.

In one sentence, this PhD thesis can be summarized as follows:

> Integrated White+Black Box Approach, a collaboration between human (the algorithm designer) and machine using FLST visualization and factorial design, is an effective methodology to address the SLS Design and Tuning Problem.

1.3 The Structure of this Thesis

As outlined in the mind map (see Figure 1.1), this thesis proceeds as follows:

1. This introductory chapter.
2. In Chapter 2, we elaborate the background materials: SLS as a promising way to attack COPs and the challenges that it faces.

[4] The term 'human', 'one', 'he', or 'algorithm designer' are interchangeable. But not everyone is capable to conduct this human-plus-computer optimization, thus this term is restricted to those with sufficient background knowledge, that is, those who are involved in the development of SLS for COP. Note that we use the term 'he' rather than 'he/she' to refer to third person just for simplicity.

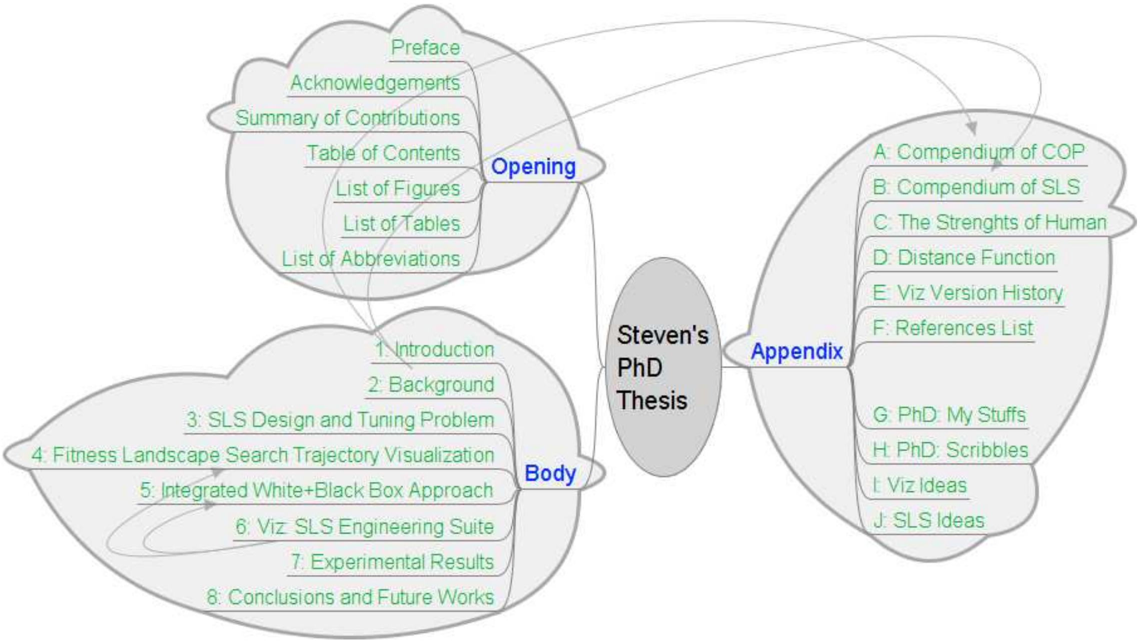

Figure 1.1: My PhD Thesis Outline

3. In Chapter 3, we present the major issue addressed in this thesis: improving the performance of SLS on various problems and situations — the SLS Design and Tuning Problem.

4. In Chapter 4, we explain the way to analyze SLS trajectory behavior on COP fitness landscape. This helps improving the overall SLS performance by matching the correct SLS strategies to the fitness landscapes of the COP at hand.

5. In Chapter 5, we summarize our overal approach for addressing the SLS Design and Tuning Problem: (1). Allow human to examine, explain SLS behavior, and tweak it (white-box) (2). Do fine-tuning using black-box approach. We called this as the Integrated White+Black Box Approach.

6. In Chapter 6, we describe the tool that we have developed: Viz. This tool consists of various visualizations to aid us in diagnosing the COP fitness landscape and SLS trajectory behavior as in Chapter 4, etc.

7. In Chapter 7, we present a series of applications of our proposed Integrated White+Black Box Approach to address the SLS Design and Tuning Problem. We tweak several SLS implementations for both well-known and new COPs.

8. In Chapter 8, we conclude our thesis and provide pointers for future direction for research along this line of work.

Several important appendices are also given:

A. List of well known COPs and some known ways to attack them.

B. List of well known SLS algorithms M including their popular configurations ϕ.

C. Short discussion of HCI.

D. List of distance functions.

E. Viz user manual and version history

Bibliography

[1] Hoong Chuin Lau, Wee Chong Wan, Steven Halim, and Kaiyang Toh. A Software Framework for Fast Prototyping of Meta-heuristics Hybridization. *International Transactions in Operational Research*, 14(2):123–141, March 2007.

[2] Hoong Chuin Lau, Wee Chong Wan, Min Kwang Lim, and Steven Halim. A Development Framework for Rapid Meta-Heuristics Hybridization. In *International Computer Software and Applications Conference*, pages 362–367, 2004.

Index