Synthesis Lectures on
Operations Research and Applications

SYNTHESIS
COLLECTION OF TECHNOLOGY

Weiqi Li

# The Traveling Salesman Problem

## Optimization with the Attractor-Based Search System

Springer

# Synthesis Lectures on Operations Research and Applications

This series focuses on the use of advanced analytics in both industry and scientific research to advance the quality of decisions and processes. Written by international experts, modern applications and methodologies are utilized to help researchers and students alike to improve their use of analytics. Classical and cutting-edge topics are presented and explored with a focus on utilization and application across a range in practical situations.

Weiqi Li

# The Traveling Salesman Problem

## Optimization with the Attractor-Based Search System

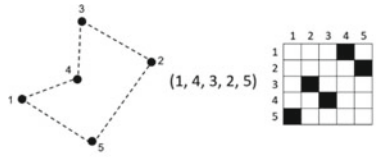
Springer

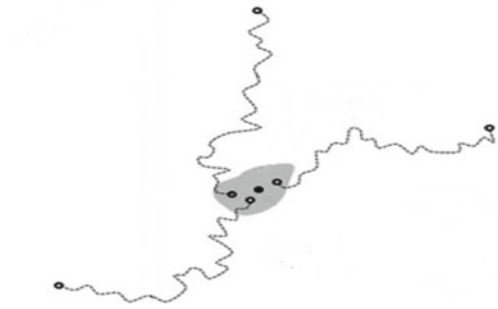
Weiqi Li
School of Management
University of Michigan
Flint, MI, USA

ISSN 2770-6303 ISSN 2770-6311 (electronic)
Synthesis Lectures on Operations Research and Applications
ISBN 978-3-031-35718-3 ISBN 978-3-031-35719-0 (eBook)
https://doi.org/10.1007/978-3-031-35719-0



This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland


3
2
4
1
5
(1, 4, 3, 2, 5)
1 2 3 4 5
1
2
3
4
5



*To Sophia and Yachen*
*for their love and support*

# Preface

The Traveling Salesman Problem (TSP) is one of the most well-known combinatorial optimization problems. Its popularity and importance can be attributed to its simple definition but high complexity to solve it, making it an ideal research problem for design of new algorithms, development of complexity theory, and analysis of search space [3–5]. The intrinsic difficulty of the TSP is associated with the combinatorial explosion of potential solutions in the solution space. There is no practically efficient algorithms to solve the TSP exactly. It can be exactly solved by only one algorithm—the brute-force (i.e., exhausted search) algorithm. Exhausted search is a trivial algorithm entailing the sequential scanning of all possibilities in the solution space. However, the time required to scan all possible solutions in the solution space increases rapidly as the size of the problem grows. Can we avoid the exhausted search? Maybe not, because we need the exhausted search to make sure the found solution is the optimal one. Then the next question is, if the exhausted search is unavoidable, can we reduce the search space to make the exhausted search feasible? If it is possible, how can we do that? This book attempts to answer these questions. The focus of this book is trying to answer the fundamental question for the TSP: "Do we ever need to explore all the solutions in the solution space to find the optimal one?" Without completely searching the solution space, how can we find the optimal solution quickly and make sure it is optimal?

Due to the intractability of the TSP, people often have to use heuristic algorithms to find reasonably good solutions to the problem. Heuristic algorithms are very efficient to find suboptimal solutions, which are very close to the optimal solution. There are many heuristic algorithms for the TSP and extensive research literature on these algorithms. This book studies the heuristic local search algorithms on the TSP from the perspective of dynamical systems. Essentially, heuristic local search algorithms for the TSP are also in the domain of dynamical systems. A dynamical system is a model of describing the temporal evolution of a system in its state space [1, 2]. Dynamical systems are called dissipative if their dynamics converge to attractors. The dynamics of dissipative systems can be thought as computation: either the initial state or parameter values can be considered the input to a computational problem, the evolution along the trajectory is the computation process, and attractor describes the solution. The goal of dynamical systems

analysis is to capture the distinctive properties of certain points in the state space of the dynamical system. The dynamical systems theory has discovered that many dynamical systems exhibit attracting behavior. That is, in such a system, all initial states tend to evolve towards a single final state or a set of states. This type of single point or a set of points in the state space is called *attractor*. The attractor theory of dynamical systems is natural paradigm that provides the necessary and sufficient theoretical foundation to study the convergent behavior of a heuristic local search system. A heuristic local search system for the TSP is a discrete dynamical system and has an attracting property that drives the search trajectories to converge to a small region in the solution space, which contains the most promising solutions to the problem. This small region is called the attractor of the local search system for the problem instance. If we can identify the attractor quickly and the attractor can be searched by an exhausted algorithm quickly, the computational complexity of the TSP can be dramatically reduced or may not exist.

The novel perspective of attractor in a local search system gives us great insights into the computational complexity of the TSP. The attractor shows us where the optimal solution can be found in the solution space. Instead of searching the entire solution space, we concentrate the exhausted search effort on this much smaller region, in which the number of possibilities is no longer prohibitive. This book presents a novel search system—the attractor-based search system (ABSS)—that can solve the TSP efficiently with optimality guarantee. The ABSS consists of two search phases: local search phase and exhausted search phase. The local search phase is to construct the attractor in which the optimal solution is located. The attractor is exponentially smaller than the solution space, and thus makes the exhausted search feasible. The exhausted search phase is to search the attractor completely to identify the optimal solution. The ABSS combines local search and exhaustes search to find the exact optimal solution quickly. Therefore, this new search paradigm is called *optimizing with attractor*.

The ABSS is designed for the TSP, not only because the TSP is an essentially important optimization problem in computational mathematics and computer science, but also because it has an exploitable data structure—the edge matrix $E$—that allows us to deal with the problem more naturally and thus provides us with a tool to significantly reduce the computational complexity of the TSP. The edge matrix $E$ helps us jump the gap from local search to global search and from stochastic search to deterministic search. In the ABSS, a local search process is for efficiency, an exhausted search process for accuracy, and the data structure matrix $E$ is the mechanism to combine them. All three elements work together to achieve the goal of the efficient and effective computation for optimization.

In fact, a heuristic local search system can work in mysterious ways. It does not have to work along the lines we think it should. The central idea of using local search process is the search space reduction, that is, the local search process can be used to reduce the number of combinatorial branching at each node. The search space reduction process removes a large number of unnecessary solutions from the search space, and guarantees

that the reduction actually remain representing the original instance but with much smaller space for the exhausted search. This book describes in detail how an attractor in a local search system drives the search trajectories into an attractor in the solution space and how the convergence of local search trajectories make the local search system become a global and deterministic system. This book also describes how to use the ABSS to solve the TSP and its variants including multi-objective TSP, probabilistic TSP, dynamic TSP, and dynamic multi-objective TSP.

This book is for general readers who are interested in optimization and/or dynamical systems. It also helps knowledgeable readers in the optimization field to deepen their understanding and initiate brainstorming. This book does not prepare to give a formal theory about the attractor of search system. Therefore, in this book, we do not intend to produce any strong mathematical proof of any theorem about the attractor and other related properties. We keep mathematics to a minimum. Instead, we describe the concepts informally based on intuition and some experimental results.

The study of new search algorithms for the TSP continues to be a vibrant, exciting and fruitful endeavor in combinatorial optimization, computational mathematics, and computer science. Numerous experts have made huge advance, but the TSP remains essentially open. A new point of view could be just what is needed to dramatically alter our ability to tackle the problem. This book proposes a new idea for improving the exhausted search. The ultimate goal of this book is to encourage readers to take up their own pursuit of interesting problem-by-problem methods for attacking diverse optimization problems. It is hoped that this book serves as a pioneer in this field and brings more and better works from other researchers and practitioners.

Flint, USA Weiqi Li

## References

1. Brin, M., Stuck, G. (2016). *Introduction to dynamical systems*. Cambridge University Press: Cambridge, UK.
2. Brown, R. J. (2018). *A modern introduction to dynamical systems*. Oxford University Press: Oxford, UK.
3. Garey, M. R., Johnson, D. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. Freemen and Company: New York.
4. Lawler, E. L., Lenstra, J. K., Rinnooy-Kan, A., Shmoys, D. (1985). *The traveling salesman problems: A guided tour of combinatorial optimization*. John Wiley & Sons: New York.
5. Papadimitriou, C. H., Steiglitz, K. (1998). *Combinatorial optimization: algorithms and complexity*. Dover Publications: New York.

# Contents


# Introduction

1

Optimization problems exist everywhere in our life. Optimization has become a critical tool in science, engineering and business. The goal of optimization is to find the best set of the admissible conditions to achieve some objectives in our decision-making process [13]. One of fundamental requirements for an optimization algorithm is to find *all optimal solutions* within a *reasonable amount of computing time*. Combinatorial optimization problems are a subset of optimization problems, which attempts to find optimal solutions from a finite set of solutions, where the set of feasible solutions is discrete or can be reduced to a discrete set. The TSP is one of the most intensively investigated combinatorial optimization problems and often treated as the prototypical combinatorial optimization problem that has provided so much motivation for design of new search algorithms, development of complexity theory, and analysis of solution space and search space [3, 19].

People have designed a variety of algorithms to solve the combinatorial optimization problems, from which two main categories can be identified: exact algorithms (i.e. the algorithms that are guaranteed to always find an optimal solution) and approximate algorithms (i.e. the algorithms that are guaranteed to find a solution that is within a certain constant factor of optimality) [3, 14, 22]. Exact search algorithms such as exhausted search, cutting-plane, branch-and-bound, and linear programming are explicitly or implicitly based on enumeration of all feasible solutions in the solution space and hence can find the exact optimal solutions, but they are very expensive from the computational perspective because they require in the worst-case an exponential number of steps. Many combinatorial optimization problems are known to be NP-complete, which means that so far there is no known asymptotically efficient algorithm that can be solve every instance of the problem in a time growing less than a power of problem size, even the seemly "limitless" increase of computer power will not help to resolve their genuine intractability [9, 12, 19]. From a computational complexity stance, intractable problems are problems for which there exist no efficient algorithms to solve them. Most intractable problems have



W. Li, *The Traveling Salesman Problem*, Synthesis Lectures on Operations Research and Applications, https://doi.org/10.1007/978-3-031-35719-0_1

an algorithm—the same algorithm—that provides an exact solution, and that algorithm is the exhausted search (also commonly called brute-force search), which is the process of examining all possible solutions in the solution space to identify the best one [8].

Due to the NP-completeness and intractability of combinatorial optimization problems, approximate approaches, based on heuristics, have become a popular means to find reasonably good solutions to these hard problem [5, 6, 20, 21, 24, 25]. Heuristics are functions that help us decide which one of a set of possible solutions is to be selected next [18]. The approximate algorithms trade in guaranteed correctness of the optimal solutions for a shorter computing time. In such an algorithm, deterministic guarantee that the optimal solution can be found is relaxed into a confidence measurement. They do not guarantee that the optimal solution will be found, instead they provide suboptimal solutions that can be very close to the optimal solution. Most heuristic search algorithms have been based on or derived from a general search technique known as heuristic local search algorithm, simply called *local search* [1].

A local search starts with an initial solution and then iteratively explores the neighborhoods of solutions trying to improve the current solution by a local change. Local search is simple to implement and quick to execute, but its search scope is limited by the neighborhood definition. As a result, it outputs a final solution that may deviate from the optimal solution. This type of final solutions is called a *locally optimal solution*, denoted by $s'$ in this book. To distinguish from a locally optimal solution, the optimal solution in the solution space is called a *globally optimal solution*, denoted by $s^*$. In order to overcome local optimality, local search usually requires some type of diversification to avoid a large region of the solution space remaining completely unexplored. The simplest way to achieve this diversification is to restart the search process from anew initial point once a solution region has been explored. The multi-start search helps explore new areas in the solution space, and therefore generates a wide sample of locally optimal points [15–17]. Multi-start algorithms can be characterized as iterative procedure consisting of two phases: (1) generating a set of random initial points and performing the local search, thus generating a set of locally optimal points, and (2) constructing adaptive starting points derived from the best locally optimal solution found so far. The entire procedure is repeated a number of iterations or certain amount of time, and the best solution is then reported. The heuristics rarely solve the problem, but they give good enough solutions based on a practical criterion.

Numerous approaches to solving the TSP have been developed. Even though the TSP is believed to be NP-complete and presumable difficult to solve exactly, with today's fast computers, we can perform an exhausted search through all the possible solutions of moderately seized instances. Modern approximate algorithms can find the solutions that get very close to the optimal solution within a few percent of optimum in a reasonable amount of time for large TSP instances with millions of nodes [4, 7, 10–12, 18, 22, 23, 26]. There are some local search algorithms for many real-world TSP instances, the empirical average-case complexity (i.e. time versus problem size) of such algorithm can