République Algérienne Démocratique et Populaire
الجمهورية الجزائرية الديمقراطية الشعبية
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
وزارة التعليم العالي والبحث العلمي

المدرسة الوطنية للإعلام الآلي
(المعهد الوطني للتكوين في الإعلام الآلي سابقا)
École nationale Supérieure d'Informatique
ex. INI (Institut National de formation en Informatique)

2nd Year of Higher Cycle

2021 - 2022

Exact and Approximate Algorithms for Combinatorial Optimisation,

The Traveling Salesman Problem as An Application

Made by :
ALIOUSALAH Mohamed Nassim
BELGOUMRI Mohammed Djameleddine
MELIANI Abdelghani
AÏT MEZIANE Mohamed Amine
LOUCIF Taha Ammar

Supervised by :
Mme. BESSEDIK M
M. KECHID A

April 28, 2022

Contents

Introduction

The traveling salesman problem (which will be denoted by TSP for brevity's sake) is a classic problem in computer science. It is a typical example of a combinatorial optimization problem, that is, an optimization problem with a *discrete* solution space.

In its simplest form, the TSP asks the following question: "A salesman wants to take *the best*[1] *possible itenerary* between a set of cities, every city must be visited exactly once, and the salesman must start and finish at the same city. How can he find this itenerary?"

It is not dificult to see the practical use of solving the TSP. In fact, many important problems like vehicle routing, scheduling, array clustering [2], and circuit design [1] can be *expressed*[2] as TSP instances.

Furthermore, the TSP is of particular theoretical interest to complexity theory researchers, as its decision variant a member of a very important family of decision problems called NP-complete problems.

In this document, we will introduce the TSP, investigate some of its properties and applications, and propose a few algorithms for solving it.

[1] Usually "best" means shortest.

[2] Formally speaking, these problems can be *reduced* to TSP.

Chapter 1

General Concepts

1.1 Computational Complexity

Throughout this document, we will often find ourselevs in need of a method to objectively measure the difficulty of a problem or the efficiency of an algorithm. Fortunately, there exists an entire branch of theoretical computer science that addresses these very questions: *the theory of computational complexity.*

A minor technical obstacle to the use of computational complexity theory on the TSP is the fact that this theory only considers so-called decision problems. The TSP beeing an optimization problem, is therefore formally speaking out of the scope of this framework.

However, this can be metigated by associating a decision problem with the TSP in such a way as to preserve its difficulty. The rest of this section will provide such an association after developing the necessary tools.

1.1.1 Combinatorial Optimization Problems

Combinatorial optimization seeks to find an *optimum* with respect to some *objective* in a discrete *space of options.* Therefore, it suffices to define these parameters in order to define a combinatorial optimization problem. Formally speakeing, we give the following definition:

Definition 1 (Optimization problem)**.**

A combinatorial optimization problem (or simply, an optimization problem) A is given by a quadruple $A := (S, R, \mu, f)$ where:

- S is a finite or countably infinite set called the *instance space* of A.
- $\forall x \in S$ $R(x)$ is the set of *feasible solutions* for the instance x.
- $\forall x \in S, \forall y \in R(x)$ $\mu(x, y) \in \mathbb{R}_+$ is the measure of y.

Using the above definition, TSP can be defined as $\mathsf{TSP} = (S, R, \mu, f)$ with:

- S the set of all complete weighted graphs.
- For $G \in S$, $R(G)$ is the set of all hamiltonian cycles in G.
- For a hamiltonian cycle C of G, $\mu(G, C)$ is the length of C.
- Finally, the problem is to minimize $\mu(G, C)$.

1.1.2 Decision Problems

In contrast to optimization problems, which can have a very large if finite space of solutions, an instance of a decision problem has a binary answer. Hence, decision problems are significantly easier to define and analyze from a theoretical point of view. Formally speaking, we define a decision problem as:

Definition 2 (Decision problem)**.**
A decision problem A is given by a pair $A := (S, B)$ where:

- S is a set called the *instance space* of A.
- $B \subset S$ is the set of *positive instances* of A (that is instances for which the answer is yes).

Chapter 2

Problem Statement

2.1 History

The first use of the term 'traveling salesman problem' in mathematical circles may have been in 1931-32, as we shall explain below. But in 1832, a book was printed in Germany entitled Der Handlungsreisende, wie er sein soil und was er zu thun hat, um Auftrage zu erhalten und eines glucklichen Erfolgs in seines Geschdften gewiss zu sein. Von einem alien Commis- Voyageur ("The Traveling Salesman, how he should be and what he should do to get Commissions and to be Successful in his Business. By a veteran Traveling Salesman').

Although devoted for the most part to other issues, the book reaches the essence of the TSP in its last chapter: 'By a proper choice and scheduling of the tour, one can often gain so much time that we have to make some suggestions.... The most important aspect is to cover as many locations as possible without visiting a location twice ...' [Voigt, 1831; MiMer-Merbach, 1983].

Chapter 3

Branch and Bound

3.1 Motivation

The direct method as we have seen is, despite the simplicity of its implementation, unrealistically slow for even very small instances.

Faster exact algorithms exist, but none of them is polynomial since TSP is NP-complete. In fact, under the assumption $P \neq NP$, no polynomial solution exists.

Branch and Bound is one such algorithm that we will dedicate the rest of the chapter to.

3.2 The idea of Branch and Bound

The idea of Branch and Bound is to eliminate certain branches from the search space to decrease runtime.

This is done by computing a *lower bound* and *upper bound* for every branch, and then pruning branches that are garanted to be worse than the best known solution.

3.3 The implementation

Bibliography

[1] Gerhard Reinelt. *The Traveling Salesman Computational Solutions for TSP Applications*. Springer, 1994.

[2] *THE TRAVELING SALESMAN PROBLEM A Guided Tour of Combinatorial Optimization*. John Wiley & Sons Ltd., 1985.