Available online at www.sciencedirect.com
SCIENCE DIRECT®
European Journal of Operational Research 160 (2005) 154–171
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/dsw

Data structures and ejection chains for solving large-scale traveling salesman problems ☆

Dorabela Gamboa [a], César Rego [b,*], Fred Glover [c]

[a] Escola Superior de Tecnologia e Gestão de Felgueiras, Instituto Politécnico do Porto, Rua do Curral, Casa do Curral, Apt. 205, 4610-156, Felgueiras, Portugal
[b] Hearin Center for Enterprise Science, School of Business Administration, University of Mississippi, MS 38677, USA
[c] Leads School of Business, University of Colorado, Boulder, CO 80309-0419, USA

Available online 8 June 2004

Abstract

Data structures play a crucial role in the efficient implementation of local search algorithms for problems that require circuit optimization in graphs. The traveling salesman problem (TSP) is the benchmark problem used in this study where two implementations of the stem-and-cycle (S&C) ejection chain algorithm are compared. The first implementation uses an Array data structure organized as a doubly linked list to represent TSP tours as well as the S&C reference structure. The second implementation considers a two-level tree structure. The motivation for this study comes from the fact that the S&C neighborhood structure usually requires subpaths to be reversed in order to preserve a feasible orientation for the resulting tour. The traditional Array structure proves to be inefficient for large-scale problems since to accomplish a path reversal it is necessary to update the predecessor and the successor of each node on the path to be reversed. Computational results performed on a set of benchmark problems up to 316,228 nodes clearly demonstrate the relative efficiency of the two-level tree data structure.

Keywords: Traveling salesman; Local search; Data structures; Ejection chains

1. Introduction

Generically, the traveling salesman problem (TSP) consists in sequentially visiting a set of clients (cities, locations) only once and finally returning to the initial client. The goal is to find the tour of minimum total distance (or other cost measure associated with the performed trajectory).

In graph theory, the problem can be defined as a graph $G = (V, A)$ with n vertices (or nodes) $V = \{v_1, \ldots, v_n\}$ and a set of edges $A = \{(v_i, v_j) | v_i, v_j \in V,\ i \neq j\}$ with a non-negative cost (or distance) matrix $C = (c_{ij})$ associated with A. The problem's resolution consists in determining the minimum cost Hamiltonian cycle on the problem graph. In this paper, we consider the symmetric version of the problem ($c_{ij} = c_{ji}$), which satisfies the triangular inequality ($c_{ij} + c_{jk} \geqslant c_{ik}$).

☆ This research has been supported in part by the Office of Naval Research (ONR) grant N000140110917.
* Corresponding author.
E-mail addresses: dgamboa@estgf.ipp.pt (D. Gamboa), crego@bus.olemiss.edu (C. Rego), fred.glover@colorado.edu (F. Glover).

0377-2217/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2004.04.023

The TSP is well known as a NP-hard combinatorial problem; hence, there is no algorithm capable to solve all possible instances in polynomial time. Consequently, it becomes absolutely necessary to use heuristic (or approximate) algorithms to provide solutions that are as good as possible but not necessarily the optimal. The importance of obtaining efficient heuristics to solve large-scale TSPs recently motivated Johnson, McGeogh, Glover, and Rego to organize the "8th DIMACS Implementation Challenge" specific for TSP algorithms [5]. This paper is based on the development of several algorithmic components and data structures with the purpose of improving the efficiency of the stem-and-cycle algorithm described in Rego [9].

A fundamental aspect forming the basis for this study concerns the following. The data structure representation of a symmetric TSP tour requires the specification of an orientation by which the tour can be read (or traversed), even though the cost of crossing in one direction or in the opposite direction is equivalent. The relevance of this orientation becomes more evident with local search algorithms where moves often require the reversal of a subpath in order to preserve an admissible orientation for a TSP tour.

A typical example of the need to reverse paths occurs in the application of classic procedures of the type k-optimal, initially proposed for the TSP [7]. The same phenomenon occurs with the moves generated in some subpath using ejection chain methods, in particular those oriented by a reference structure.

Naturally, the need to reverse paths at each iteration of the algorithm requires a computational effort that becomes particularly relevant when large-scale problems have to be solved.

The choice of the data structure to represent TSP solutions is crucial when applying neighborhood structures that require path reversals, since the algorithm's complexity might drastically be reduced. Fredman et al. [2] show the relevance of that choice on their implementation of the Lin–Kernighan algorithm [8] by comparing the computational times obtained by several implementations using four different data structures: Array, splay-tree, two-level tree and segment tree.

The main goal of this study consists in analyzing and validating the potential of the two-level tree data structure in reducing the running time of the stem-and-cycle algorithm [9] with the aim of improving the algorithm's efficiency in solving large-scale problems.

The motivation for this study comes from the fact that the stem-and-cycle algorithm has proved to be extremely effective and competitive with the best algorithms for the TSP [3,6]. We therefore anticipated that the algorithm's efficiency when solving large scale problems can be improved by the implementation of a data structure specifically designed to reduce the computational complexity associated with the path reversal operations needed at each iteration of the algorithm.

2. Data structures for the S&C procedure

2.1. The stem-and-cycle reference structure

The stem-and-cycle (S&C) reference structure is described in Glover [4] and used in the subpath ejection algorithm described in Rego [9] for the TSP.

In graph theory, the S&C structure is defined by a spanning subgraph of G, consisting of a path $\mathrm{ST} = (v_t, \ldots, v_r)$ called the stem, attached to a cycle $\mathrm{CY} = (v_r, v_{s_1}, \ldots, v_{s_2}, v_r)$. Vertex v_r in common to the stem and the cycle is called the root and, consequently, its adjacent vertices v_{s_1} and v_{s_2} are called subroots. Finally vertex v_t is called the *tip* of the stem. Fig. 1 shows a representation of the stem-and-cycle structure.

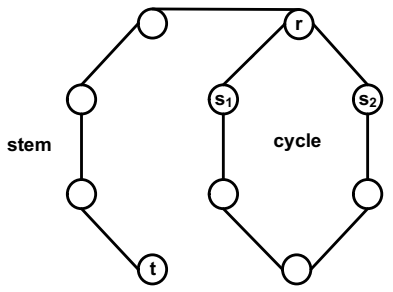

Fig. 1. The stem-and-cycle reference structure.