European Journal of Operational Research 235 (2014) 115–128

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Exact algorithms for the traveling salesman problem with draft limits

Maria Battarra [a,*], Artur Alves Pessoa [b], Anand Subramanian [c], Eduardo Uchoa [b]

[a] University of Southampton, School of Mathematics, Southampton SO17 1BJ, UK
[b] Universidade Federal Fluminense, Departamento de Engenharia de Produção, Rua Passo da Pátria 156, Bloco E, 4° andar, São Domingos, Niterói, RJ 24210-240, Brazil
[c] Universidade Federal da Paraíba, Departamento de Engenharia de Produção, Centro de Tecnologia, Campus I, Bloco G, Cidade Universitária, João Pessoa, PB 58051-970, Brazil

ARTICLE INFO

Article history:
Received 20 February 2013
Accepted 15 October 2013
Available online 28 October 2013

Keywords:
Draft limits
Traveling salesman
Cutting planes
Column generation
Extended formulation

ABSTRACT

This paper deals with the Traveling Salesman Problem (TSP) with Draft Limits (TSPDL), which is a variant of the well-known TSP in the context of maritime transportation. In this recently proposed problem, draft limits are imposed due to restrictions on the port infrastructures. Exact algorithms based on three mathematical formulations are proposed and their performance compared through extensive computational experiments. Optimal solutions are reported for open instances of benchmark problems available in the literature.

1. Introduction

The Traveling Salesman Problem (TSP) with Draft Limits (TSPDL) is a variant of the well-known TSP, recently introduced by Rakke, Christiansen, Fagerholt, and Laporte (2012), that arises in the context of maritime transportation. The sequence of ports that a cargo ship visits in a tour is dependent on the port infrastructures: the sea-level in a port is sometimes not sufficiently deep to accommodate loaded cargo ships. The port is thus associated to a draft limit, i.e., the maximum vertical distance allowed between the waterline and the bottom of the hull. Note that the draft of a cargo ship depends on the load: the heavier the load, the higher the ship's draft. Therefore, draft limits can be easily translated into restrictions on the maximum load of the ship.

The problem can be formalized as follows: A directed graph $G = (V, A)$ is given, where $V = \{0, \ldots, n\}$ is the set of ports to be visited and $A = \{(i,j), i,j \in V, i \neq j\}$ is the arc set, or set of connections between ports. Each arc $(i,j) \in A$ is associated to a routing cost $c_{ij} > 0$. The vertex 0 is the port from which the ship starts and ends its tour, whereas vertices $V' = \{1, \ldots, n\}$ are ports to be visited exactly once. Each port requires the delivery of $d_i,\ i \in V'$, units of load and is associated to a draft limit $l_i,\ i \in V'$. The initial load is $Q = \sum_{i \in V'} d_i$ and we denote $\underline{d} = \min_{i \in V'}\{d_i\}$. The ship cannot enter port i if its load is heavier than l_i, or the hull of the ship could be grounded. Therefore the TSPDL asks for the minimum cost Hamiltonian tour, visiting each port exactly once and not violating draft limit constraints.

Despite its simple definition, the TSPDL proves hard to solve to optimality. In fact, the problem is $\mathcal{NP}$-Hard since it includes the TSP as special case when the drafts are sufficiently large.

Rakke et al. (2012) proposed two mathematical formulations: the first formulation makes use of the binary variables x_{ij} assuming value 1 if arc (i,j) is in the solution, and continuous variables y_{ij}, representing the load on the ship while traveling arc (i,j). The resulting formulation is compact, but provides poor quality bounds. The second formulation includes two additional sets of variables: u_j and t_{ij} specifying the position of port j and of arc (i,j) in the circuit, respectively. These two sets of variables allow for the introduction in the model of the Miller, Tucker, and Zemlin (MTZ) constraints (Miller, Tucker, & Zemlin, 1960). The MTZ constraints, usually employed to avoid subtours, are used to strengthen the formulation and they are included at the root node of the branch-and-bound tree.

Both formulations have been further strengthened by dynamically separating subtour elimination constraints (Dantzig, Fulkerson, & Johnson, 1954), as well as their lifted counterpart (Balas & Fischetti, 2004). Moreover, lower bounds on the u_j variables and lower bounds on the sum of y_{ij} variables are imposed.

The branch-and-cut algorithms originating from both formulations are capable of solving quite effectively instances with a limited amount of ports with draft limits, but when the percentage of ports with a draft limit increases, the algorithm struggles even for medium-sized instances. The problem seems therefore challenging and, as far as we are aware, no other attempts have been made to solve it exactly. This motivated our interest in the problem and we

* Corresponding author. Tel.: +44 (0)23 8059 3863.
E-mail addresses: m.battarra@soton.ac.uk (M. Battarra), artur@producao.uff.br (A.A. Pessoa), anand@ct.ufpb.br (A. Subramanian), uchoa@producao.uff.br (E. Uchoa).

0377-2217/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2013.10.042

decided to investigate alternative formulations and solution techniques.

Three alternative mathematical formulations are introduced (see Section 2). The first formulation is based on two-index variables, the second formulation is based on three-index variables, whereas the third can be viewed as an improvement over a Dantzig–Wolfe decomposition of the second, including the concept of *ng*-paths (Baldacci, Mingozzi, & Roberti, 2011), and it is solved through a branch-cut-and-price algorithm.

In Section 3, a description of the branch-and-cut and branch-cut-and-price implementations are presented. The results of our computational experience are summarized in Section 4, while Section 5 presents conclusions and future possible research directions.

2. Mathematical formulations

The exact algorithms proposed in this work are based on three integer programming formulations that are described in this section.

2.1. Formulation F1

The first formulation, denoted F1, is based on two sets of two-indexed binary variables. The x_{ij} variable assumes value 1 if $(i,j) \in A$ is in the solution, 0 otherwise. The variable y_{ik} assumes value 1 when the ship enters port $i \in V'$ carrying $k \in \{d_i, \ldots, l_i\}$ units of load, 0 otherwise. Note that arcs $(i,j)|d_j > l_i - d_i$ can be removed from the network: in order to simplify the notation we do not explicitly remove these arcs, but it is sufficient to disregard the corresponding variables in our models to take this aspect into account. The formulation F1 can be stated as follows:

$$\text{(F1)} \quad \min \sum_{(i,j)\in A} c_{ij} x_{ij} \tag{1}$$

$$\text{s.t.} \sum_{i\in V, i\neq j} x_{ij} = 1, \quad \forall j \in V \tag{2}$$

$$\sum_{j\in V, j\neq i} x_{ij} = 1, \quad \forall i \in V \tag{3}$$

$$\sum_{i\in V'|l_i \geqslant k} y_{ik} \leqslant 1, \quad \forall k \in \{\underline{d}, \ldots, Q - \underline{d}\} \tag{4}$$

$$\sum_{k=d_i}^{l_i} y_{ik} = 1, \quad \forall i \in V' \tag{5}$$

$$y_{iQ} = x_{0i}, \quad \forall i \in V'|l_i = Q \tag{6}$$

$$y_{id_i} = x_{i0}, \quad \forall i \in V' \tag{7}$$

$$x_{ij} + y_{ik} \leqslant 1 + y_{j,k-d_i}, \quad \forall (i,j) \in A, \; k \in \{d_j + d_i, \ldots, \min\{l_i, l_j + d_i\}\} \tag{8}$$

$$x_{ij} \in \{0,1\}, \quad \forall (i,j) \in A \tag{9}$$

$$y_{ik} \in \{0,1\}. \quad \forall i \in V', k \in \{d_i, \ldots, l_i\} \tag{10}$$

The Objective Function (1) aims at minimizing routing costs. Constraints (2) and (3) are the degree constraints. Constraints (4) impose that the ship visits at most a port for each intermediate load value. Constraints (4) are not necessary to define an optimal integer solution, because Constraints (2), (3), (6) and (7) guarantee that the ship performs a Hamiltonian tour in which the load is monotonically decreasing. Preliminary experiments showed that these constraints strengthen the linear relaxation and we included them in the formulation. Constraints (5) state that each port has to be assigned to a load. The first and last position of the tour are imposed to be connected to the initial port 0 (Constraints (6) and (7), respectively). Constraints (8) link variables x_{ij} and y_{ik}: if arc (i,j) is traversed, $x_{ij} = 1$ and i and j are located in consecutive positions of the tour. Therefore summing variables x_{ij} and y_{ik} can result in a value equal to 2 only if $y_{j,k-d_i} = 1$. These constraints generalize similar constrains encountered in single-machine scheduling problems (an interested reader can refer to the models based on assignment and positional date variables in Keha, Khowala, & Fowler, 2009). Finally, Constraints (9) and (10) define the binary nature of the variables. Note that Constraints (5)–(8) ensure that the flow is monotonically decreasing along the tour and therefore subtours are avoided.

Formulation F1 presents similarities with the MTZ-based formulation for the *Asymmetric TSP* (ATSP) (see Roberti & Toth, 2012 for a recent overview and comparison of ATSP models), but consists only of binary variables.

F1 is strengthened by incorporating the trivial constraints

$$x_{ij} + x_{ji} \leq 1, \quad \forall (i,j) \in A \tag{11}$$

and by separating in a cutting plane fashion the subtour elimination constraints:

$$\sum_{i\in S} \sum_{j\in \bar{S}} x_{ij} \geqslant 1. \quad \forall S \subseteq V' \tag{12}$$

For this latter set of inequalities, the exact separation can be done in polynomial time.

2.2. Formulation F2

Formulation F2 considers three-indexed binary variables z_{ij}^k, assuming value 1 if arc $(i,j) \in A$ is traversed by the ship carrying k units of load (including the demand of port j). By denoting $K_{ij} = \min\{l_j, l_i - d_i\}$, formulation F2 is:

$$\text{(F2)} \quad \min \sum_{(i,j)\in A} \sum_{k=d_j}^{K_{ij}} c_{ij} z_{ij}^k \tag{13}$$

$$\text{s.t.} \sum_{\substack{i\in V \\ i\neq j}} \sum_{k=d_j}^{K_{ij}} z_{ij}^k = 1, \quad \forall j \in V \tag{14}$$

$$\sum_{\substack{j\in V|j\neq i \\ l_j \geqslant k+d_j}} z_{ji}^k - \sum_{\substack{j\in V|j\neq i \\ l_j \geqslant k-d_i, d_j \leqslant k-d_i}} z_{ij}^{k-d_i} = 0, \quad \forall i \in V, \; k \in \{d_i, \ldots, l_i\} \tag{15}$$

$$z_{i0}^k = 0, \quad \forall i \in V', \; k \in \{1, \ldots, Q\} \tag{16}$$

$$z_{0j}^k = 0, \quad \forall j \in V'|l_j = Q, \; k < Q, \; k \in \{d_j, \ldots, l_j\} \tag{17}$$

$$z_{ij}^k \in \{0,1\}. \quad \forall (i,j) \in A, \; k \in \{d_j, \ldots, K_{ij}\} \tag{18}$$

The Objective Function (13) minimizes routing costs. Constraints (14) are the degree constraints. Constraints (15) preserve the load conservation. Constraints (16) and (17) force the ship to return to the depot empty and leave the depot carrying Q units, respectively. Constraints (18) define the nature of the variables. Formulation F2 is similar to the three-index formulations proposed in Fox, Gavish, and Graves (1980), for the *Time Dependent TSP*.

Constraints

$$\sum_{\substack{j\in V \\ j\neq i}} \sum_{k=d_j}^{K_{ij}} z_{ij}^k = 1, \quad \forall i \in V \tag{19}$$

are implied by Constraints (14) and Constraints (15), as previously stated in Pessoa, Poggi de Aragão, and Uchoa (2008).

F2 can be strengthened by the trivial constraints:

$$\sum_{k=d_j}^{K_{ij}} z_{ij}^k + \sum_{k=d_i}^{K_{ji}} z_{ji}^k \leqslant 1, \quad \forall (i,j) \in A \tag{20}$$

that have been included in the formulation *a priori*.

Flow conservation constraints ensure that subtours are avoided for F2 integer solutions, however we strengthened the formulation by including the subtour elimination constraints as cutting planes (as for F1):