Computers & Operations Research 40 (2013) 3010–3020

Contents lists available at ScienceDirect

Computers & Operations Research

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

Branch and bound algorithms for the bus evacuation problem ☆

Marc Goerigk *, Bob Grün, Philipp Heßler

Fachbereich Mathematik, Technische Universität Kaiserslautern, Germany

ARTICLE INFO

Available online 17 July 2013

Keywords:
Disaster management
Integer programming
Evacuation planning
Branch and bound
Heuristics

ABSTRACT

The bus evacuation problem (BEP) is a vehicle routing problem that arises in emergency planning. It models the evacuation of a region from a set of collection points to a set of capacitated shelters with the help of buses, minimizing the time needed to bring the last person out of the endangered region.

In this work, we describe multiple approaches for finding both lower and upper bounds for the BEP, and apply them in a branch and bound framework. Several node pruning techniques and branching rules are discussed. In computational experiments, we show that solution times of our approach are significantly improved compared to a commercial integer programming solver.

1. Introduction

Recent events like hurricanes over North America (see [7]), or tsunamis in the Indian Ocean remind us that evacuating whole regions may become necessary in case of an emergency; and, in such a situation, operations research is able to save both lives and expenses. For a survey on models and challenges in this area of research, see, e.g., Hamacher and Tjandra [5], Altay and Green [1], or Yusoff et al. [10].

In this work, we consider the problem of evacuating an urban region to a set of emergency shelter locations with the help of available public transport infrastructure. In particular, we assume that evacuees that do not travel on their own, be it due to age, sickness, the lack of a private car or any other reason, are gathered at few collection points, where they are brought on buses. The arising optimization problem is to determine a set of bus routes along with their timetable that minimizes the evacuation time, i.e., the time needed for the last evacuee to reach a shelter.

The problem we consider is closely related to the one discussed in Bish [2], where mixed-integer programming models and an iterative local search heuristic are presented. Making use of public transport in emergency evacuation is also considered in Sayyady and Eksioglu [9], incorporating traffic flow dynamics. Recently, Goerigk and Grün [4] considered the uncertain bus evacuation problem, where the exact number of evacuees is not available at the beginning of the evacuation planning.

Contributions. Naturally, planning in emergency situations has strict computation time limitations, and the usage of sub-optimal heuristics stands to reason. We discuss several approaches to construct feasible solutions to the bus evacuation problem, and to calculate lower bounds on the evacuation time that helps the planner to assess the situation.

However, being useful on their own, we show that these upper and lower bounds can be easily integrated into a branch and bound framework that aims at finding an optimal solution. In the computational experiments we show that such an algorithm can solve the most realistically sized instances in reasonable time; furthermore, the resulting computation times are significantly smaller than when a commercial IP solver is used.

Overview. In Section 2 we describe the bus evacuation problem in detail. We then present upper bounds for the problem in Section 3, and lower bounds in Section 4. In Section 5, we discuss branching rules and node reduction techniques. Computational results are presented in Section 6. Finally, Section 7 concludes the paper.

2. Problem description

In this section we formalize the problem we consider, and model it with the help of a linear integer programming formulation.

The evacuation scenario we consider is the following: a densely populated region needs to be evacuated, and not everybody is able to leave the region on its own. An example for such a situation is the defusal of a bomb within a city center, as is frequently necessary in cities bombed during World War II. Emergency shelters are prepared for the evacuees, and buses of the local transport agency are used.

The collection points $[S] = \{1, \ldots, S\}$ where evacuees gather will be referred to as *sources*, and the shelters $[T] = \{1, \ldots, T\}$ where they

☆Partially supported by the Federal Ministry of Education and Research Germany, Grant DSS_Evac_Logistic, FKZ 13N12229; and by the European Union Seventh Framework Programme (FP7-PEOPLE-2009-IRSES) under grant agreement no. 246647 and by the New Zealand Government as part of the OptALI project.

* Corresponding author. Tel.: +49 631 205 3878; fax: +49 631 205 4737.
E-mail addresses: goerigk@mathematik.uni-kl.de (M. Goerigk), bgruen@mathematik.uni-kl.de (B. Grün), hessler@mathematik.uni-kl.de (P. Heßler).

0305-0548/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2013.07.006

are transported to as *sinks*. We assume that the number of evacuees at every source $i\in[S]$ is known, and given in terms of integer multiples of bus loads denoted by l_i. Furthermore, every sink $j\in[T]$ can only shelter a limited number of evacuees, and is thus given a capacity u_j. We will refer to the number of evacuees and the shelter capacities as *lower* and *upper* bounds, or as *supply* and *demand*. We denote the total number of evacuees with $L=\sum_{i\in[S]} l_i$. All buses start from a depot that is not necessarily a source or a sink. Formally, the bus evacuation problem (BEP) is now given as

The bus evacuation problem (*BEP*):

Input: The number of buses B, of sources S, and of sinks T. A matrix $(d_{ij})_{i\in[S],j\in[T]}$ of source–sink–distances, a vector $(d_i^{start})_{i\in[S]}$ of depot–source–distances, a vector $(l_i)_{i\in[S]}$ of numbers of evacuees, and a vector $(u_j)_{j\in[T]}$ of sink capacities.

Find: Find a bus schedule minimizing the maximum travel time over all buses such that all evacuees are transported to the sinks, and sink capacities are respected.

The problem was first described in Bish [2], where both sources and sinks are nodes in a transportation network, and buses are allowed to pick up and drop off amounts of evacuees up to a given bus capacity. Here we consider a simplified problem version; however, all described algorithms can also be applied to the problem definition of Bish [2]. We explain the problem using a small example instance.

Example 1. Fig. 1 illustrates a BEP instance. There are three sources and three sinks given. The sources have a supply of $l=(1,\ 3,\ 3)$, while the sinks have capacities of $u=(4,\ 4,\ 1)$. The distance from the depot to the source is given by $d^{start}=(7,\ 4,\ 9)$, and the distances between S and T are given by

$$d=\begin{pmatrix} 6 & 7 & 8 \\ 10 & 9 & 2 \\ 6 & 3 & 7 \end{pmatrix}.$$

The number of buses is $B=3$.

Table 1 represents a feasible solution to the presented instance; in fact, it is even optimal. The first bus travels from source 1 to sink 1, and then from source 3 to sink 2. Its total travel time is thus given by $d_1^{start}+d_{11}+d_{31}+d_{32}=7+6+6+3=22$. For bus 2 we calculate a travel time of 23, and for bus 3 a travel time of 23 again, resulting in a total evacuation time of 23.

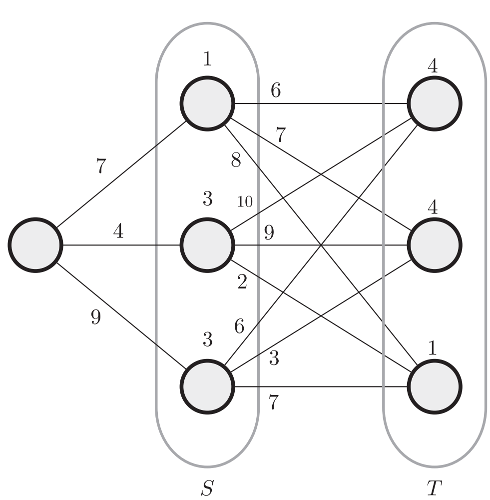

Fig. 1. Example for a BEP instance.

Table 1
Feasible solution.

Trip nr.	1	2	3
Bus 1	(1, 1)	(3, 2)	–
Bus 2	(2, 1)	(3, 2)	–
Bus 3	(2, 3)	(2, 2)	(3, 2)

We shall refer to a pair (i,j) of source and sink node as a *tour*, and to a list of tours as a *tourplan*.

In order to model the problem as a mixed-integer linear program, we fix a maximum number of rounds R the evacuation process might possibly take (a trivial upper bound on R is given by $\sum_{i\in[S]} l_i$). We introduce variables $x_{ij}^{br}\in\mathbb{B}$ that represent whether bus b travels from source i to sink j in round r. The variables t_{to}^{br} and t_{back}^{br} measure the travel time for bus b in round r from the source to the sink, and from the sink to the next source, respectively. Finally, the variable t_{max} denotes the maximum total travel distance over all buses:

$$\min \quad t_{max} \tag{1}$$

$$\text{s.t.}\quad t_{max}\geq \sum_{r\in[R]} (t_{to}^{br}+t_{back}^{br})+\sum_{i\in[S]}\sum_{j\in[T]} d_i^{start} x_{ij}^{b1} \quad \forall b\in[B]. \tag{2}$$

$$t_{to}^{br}=\sum_{i\in[S]}\sum_{j\in[T]} d_{ij}x_{ij}^{br} \quad \forall b\in[B],\ \ r\in[R] \tag{3}$$

$$t_{back}^{br}\geq d_{ij}\left(\sum_{k\in[S]} x_{kj}^{br}+\sum_{l\in[T]} x_{il}^{b,r+1}-1\right) \tag{4}$$

$$\forall b\in[B],\ \ r\in[R-1],\ \ i\in[S],\ \ j\in[T] \qquad \sum_{i\in[S]}\sum_{j\in[T]} x_{ij}^{br}\leq 1 \quad \forall b\in[B],\ \ r\in[R] \tag{5}$$

$$\sum_{i\in[S]}\sum_{j\in[T]} x_{ij}^{br}\geq \sum_{i\in[S]}\sum_{j\in[T]} x_{ij}^{b,r+1} \quad \forall b\in[B],\ \ r\in[R-1] \tag{6}$$

$$\sum_{j\in[T]}\sum_{b\in[B]}\sum_{r\in[R]} x_{ij}^{br}\geq l_i \quad \forall i\in[S] \tag{7}$$

$$\sum_{i\in[S]}\sum_{b\in[B]}\sum_{r\in[R]} x_{ij}^{br}\leq u_j \quad \forall j\in[T] \tag{8}$$

$$x_{ij}^{br}\in\mathbb{B} \quad \forall i\in[S],\ \ j\in[T],\ \ b\in[B],\ \ r\in[R] \tag{9}$$

$$t_{to}^{br},t_{back}^{br}\in\mathbb{R} \quad \forall b\in[B],\ \ r\in[R] \tag{10}$$

$$t_{max}\in\mathbb{R} \tag{11}$$

Constraint (2) ensures that t_{max} is as large as the maximal travel time of all buses. Constraints (3) and (4) are used to measure the travel time, while Constraint (5) makes sure that a bus can only travel from one source to one sink per round. Constraint (6) models that bus tours are connected and can stop whenever they like, while Constraints (7) and (8) ensure that all evacuees are transported and shelter capacities are respected.

It is shown in Goerigk and Grün [4] that BEP is NP-complete, even in the case of $d_i^{start}=0$ and $d_{ij}=d_{i'j}$ for all $i,\ i'\in S$.

3. Upper bounds: algorithms for constructing feasible solutions

We now develop four greedy approaches to construct a feasible solution to the BEP. All algorithms are able to make use of solutions that are partially fixed, which will be of importance for the usage within a branch and bound framework. In particular, we may assume