Computers & Operations Research 39 (2012) 964–984

Contents lists available at ScienceDirect

Computers & Operations Research

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

Solving shortest path problems with a weight constraint and replenishment arcs

Olivia J. Smith [a,*], Natashia Boland [b], Hamish Waterer [b]

[a] University of Melbourne, Department of Mathematics and Statistics, Australia
[b] University of Newcastle, School of Mathematical and Physical Sciences, Australia

ARTICLE INFO

Available online 22 July 2011

Keywords:
Resource constraints
Shortest paths
Preprocessing
Replenishment
Labeling algorithms

ABSTRACT

This paper tackles a generalization of the weight constrained shortest path problem (WCSPP) in a directed network with *replenishment arcs* that reset the accumulated weight along the path to zero. Such situations arise, for example, in airline crew pairing applications, where the weight represents duty hours, and replenishment arcs represent crew overnight rests; and also in aircraft routing, where the weight represents time elapsed, or flight time, and replenishment arcs represent maintenance events. In this paper, we review the weight constrained shortest path problem with replenishment (WCSPP-R), develop preprocessing methods, extend existing WCSPP algorithms, and present new algorithms that exploit the inter-replenishment path structure. We present the results of computational experiments investigating the benefits of preprocessing and comparing several variants of each algorithm, on both randomly generated data, and data derived from airline crew scheduling applications.

1. Introduction

Given a directed network, together with a start node, an end node, and a cost and a non-negative weight value for each arc, the *weight constrained shortest path problem* (*WCSPP*) is the problem of finding a least cost path in the network from the start node to the end node, subject to a limit on the total weight. This paper extends the WCSPP to include *replenishment arcs* which reset weight accumulation to zero, giving the *weight constrained shortest path problem with replenishment* (*WCSPP-R*). In this problem, the total accumulation of weight at any point in a feasible path cannot exceed the weight limit.

The WCSPP has been widely studied because of its relevance to important practical applications, such as crew scheduling, rostering, aircraft routing and telecommunications. Interestingly, most of these applications exhibit replenishment opportunities. In crew scheduling and rostering, rest periods, for example, either overnight in the case of scheduling, or for periods of two or more days in the case of rostering, replenish crews' ability to work [10–13,19,20,27,32,33,35–37]. In aircraft routing, maintenance events replenish aircrafts' ability to fly [1,10,11,32,33,38]. In telecommunications, equipment can be placed in the network to replenish a signal [7,8]. Many of these applications are addressed via column generation techniques, which often lead to constrained shortest path subproblems. We discuss this point in more detail later.

For now, we observe that the majority of work that tackles replenishment directly has been in the context of the traveling salesman problem (TSP) variants, in which the salesman cannot visit too many nodes in a row, or travel too far, without visiting one of a limited number of "replenishment" nodes or arcs [3,5,6,22,26,29–31]. The asymmetric traveling salesman problem with replenishment arcs (RATSP) was introduced in the PhD thesis of Zhu [38]. It was later developed by Boland et al. [5] who provided a column generation formulation where each column represents an inter-replenishment path. Heuristic methods for the RATSP are given by Mak and Boland [29], its polyhedral structure is investigated in [30], and a Lagrangian relaxation based branch-and-bound algorithm in which strong cutting planes are relaxed is developed in [31]. A more recent line of work on replenishment in the context of the TSP has been based around the black and white traveling salesman problem (BWTSP) which uses replenishment nodes and limits the number of these in any feasible solution. Bourgeois et al. [6] solve this problem using construction, feasibility restoration and improvement (2-opt) heuristics.

* Corresponding author.
E-mail addresses: o.smith@ms.unimelb.edu.au, livsmith21@gmail.com (O.J. Smith).

0305-0548/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2011.07.017

Ghiani et al. [22] provide cuts and an exact branch-and-cut method for an arc-based formulation of this problem. The heuristic of Bhattacharya et al. [3] provides a known bound on the BWTSP. Jiang et al. [26] provide a polynomially sized formulation of the BWTSP without computational results. Whilst the work on the TSP with replenishment provides interesting background, we expect quite different methods to be successful in the shortest path context.

As mentioned above, important applications, such as crew scheduling and aircraft routing, exhibit replenishment characteristics. For example, Barnhart et al. [2] describe resources such as flying time, elapsed time, and number of flights in the current duty, all of which are replenished by an inter-duty rest. In many cases, the solution approach exploits the replenishment structure by having column generation master problem variables represent inter-replenishment activity sequences. For aircraft routing, flight "string" models, in which master problem variables are flight sequences between maintenance opportunities, are prevalent, appearing, for example, in the work of Barnhart et al. [1] and Cohn and Barnhart [10]. Mercier et al. [32] and Mercier and Soumis [33], in treating aircraft routing, expand their master problem to cover d copies of each flight for each of the d days that can elapse between maintenance opportunities. In either of these approaches, the end effect is to embed replenishment structure in the master problem, and avoid the need to model replenishment in the subproblem. For airline crew pairing, similar approaches and the use of duty networks, for example in Klabjan et al. [27] and Vance et al. [36], also avoid replenishment in the subproblem. When the specifics of crew rules and operating circumstances permit, duties can be enumerated and inter-duty replenishment requirements embedded in the subproblem network's arc structure, as in Yan [37]. For airline crew rostering, Day and Ryan [12] also avoid replenishment by first computing days off patterns; in the second stage (solved by a restricted enumeration approach), work sequences are inserted.

Other column generation approaches do appear to have subproblems that could be represented as constrained shortest path problems having some form of replenishment. The work of Falkner and Ryan [20] on bus driver scheduling is one such paper, where bus drivers require meal breaks. The airline crew scheduling [32,33] would appear to use flight networks for the subproblem, and it could be expected that inter-duty replenishment requirements would arise in their subproblem. In earlier work, Cordeau et al. [11], who also address integrated airline crew scheduling and aircraft routing, have resource constrained shortest path problems for both their aircraft and crew subproblems. Ziarati et al. [39] schedule locomotives into "consists" (groups of locomotives) over a weekly dated schedule, where some critical locomotives require maintenance, a form of replenishment provided by the use of "shop arcs" in the time–space network. Fahle et al. [19] and Sellmann et al. [35] solve crew rostering problems by column generation. They wish to consider a large class of complex rules, which they believe are not readily modeled by constrained shortest paths, and so take a constraint programming approach to solving the subproblem. However they do use some standard shortest path preprocessing techniques within their constraint programming method. These papers – with the exception of [19,35] – provide unfortunately very little detail about either the precise form of the subproblem, or the method used to solve it. Sometimes the subproblems appear to be solved by enumeration, or a restricted enumeration heuristic. More usually there is mention of dynamic programming or labeling methods in some form being used, with references to sources such as the book chapters of Desaulniers et al. [14] or Desrosiers et al. [16]. However the more recent book chapter of Irnich and Desaulniers [25] provides an excellent and detailed description of the variety of shortest path subproblems that can arise in column generation. An overview of solution techniques is also given, and work providing new methodological advances surveyed. This work highlights the importance of constrained shortest path problems in general, and the need for more efficient algorithms for both specific variants and more general problems.

Before addressing the WCSPP-R, we observe that the WCSPP has received a significant amount of attention in the literature. Starting with the seminal work of Desrochers and Soumis [15], most approaches are based on some form of labeling algorithm. Significant work has highlighted the utility of preprocessing procedures, and the use of Lagrangian relaxation, enumeration, kth-shortest paths, and combinations of these ideas to achieve substantial computational improvements. The work of Zieglemann [40] was among the first to address some of these combinations. The most recent sequence of work combines preprocessing and Lagrangian relaxation with label setting [17], interleaves Lagrangian relaxation and preprocessing, combined with enumeration [34], and culminates with Carlyle et al. [9], which integrates Lagrangian relaxation, enumeration and preprocessing for problems with one or more weight constraints, to find ε-optimal solutions with controllable ε. The latter also includes a broader overview of methods for WCSPP.

We now turn our attention to the specific problem of the WCSPP-R. Like the WCSPP, the WCSPP-R is NP-Hard. The former is shown in Garey and Johnson [21]. The latter is easily obtained by noting that the WCSPP is a special case of WCSPP-R. Also like the WCSPP, the WCSPP-R can be solved in pseudo-polynomial time. This follows from results of Cabral [18] on a related problem.

By contrast with the WCSPP, there has been relatively little literature to date addressing the WCSPP-R, or close relatives, of it. It can be viewed as a special case of the general resource constrained shortest path framework provided by Irnich [24]. The WCSPP-R additive resource with replenishment can be represented as an example of the general *resource extension function* (REF) presented in [24]. Irnich [24] does not go deeply into specific algorithms, but does give properties of REFs that allow important algorithmic procedures. The WCSPP-R satisfies in particular the "inversion" property, which is relevant to our preprocessing procedure; we give more detail in the corresponding section. The WCSPP-R can also be modeled as a shortest path problem with time windows (SPPTW), by setting arc travel times to be the arc weights on non-replenishment arcs and the negative of the weight limit on replenishment arcs, the lower time window bound at zero and the upper time window bound at the weight limit for all nodes. However the use of negative arc weights prohibits useful preprocessing procedures in other than acyclic graphs, and the rather loose time windows obscure the structure of the problem. Addressing this structure so as to arrive at efficient methods is the main focus of our paper.

The only other work we are aware of which specifically considers replenishment in the context of the WCSPP is the PhD thesis of Cabral [18] and the paper of Laporte and Pascoal [28]. This work considers the *shortest path problem with relays* (*SPPR*), which deals with an undirected network in which replenishment can occur at any node, at the price of some node-dependent cost. As in the WCSPP-R, in the SPPR a path can accumulate no more than a given weight limit before replenishment must occur. Cabral [18] presents three algorithms for the SPPR. Two are label correcting algorithms that use different methods of storing the labels. The third uses the structure of a feasible path, which consists of a sequence of subpaths between replenishments, to develop a method in which a shortest path is found in a "higher level" network. In this network, replenishment occurs at all nodes, and arcs represent the minimum cost weight-feasible replenishment-free path between the two nodes in the original network. This latter algorithm is found to be far less efficient than either of the label correcting methods. Laporte and Pascoal [28] present both a label setting and a label correcting