Computers & Operations Research 40 (2013) 385–394

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

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

Implementation of a three-stage approach for the dynamic resource-constrained shortest-path sub-problem in branch-and-price

Xiaoyan Zhu [a,*], Wilbert E. Wilhelm [b]

[a] Industrial and Information Engineering Department, University of Tennessee, 412 East Stadium Hall, Knoxville, TN 37996-0700, USA
[b] Industrial and System Engineering Department, Texas A&M University, TAMUS 3131, College Station, TX 77843-3131, USA

ARTICLE INFO

Available online 13 July 2012

Keywords:
Resource-constrained shortest-path problem
Three-stage approach
Re-optimization
Fixed arcs
Column generation
Branch-and-price

ABSTRACT

The resource-constrained shortest-path problem (RCSP) is often used as a sub-problem in branch-and-price because it can model the complex logic by which many actual systems operate. This paper addresses two special issues that arise in such an application. First, RCSP in this context is dynamic in the sense that arc costs are updated at each column-generation iteration, but constraints are not changed. Often, only a few arc costs are updated at an iteration. Second, RCSP must be solved subject to arcs that are forbidden or prescribed as corresponding binary variables are fixed to 0 or 1 by the branching rule. A reoptimizing algorithm for dealing with a few arc-cost changes and a method for dealing with fixed arcs are proposed and incorporated into a three-stage approach, specializing it for repeatedly solving the dynamic RCSP as a sub-problem in branch-and-price. Computational tests evaluate the effectiveness of the proposed algorithms.

1. Introduction

The resource-constrained shortest-path problem (RCSP) is a variant of the shortest-path problem (SPP). Let $\Re$ be a set of non-negative, discrete-valued resources and $T=(T_1,\ldots,T_{|\Re|})$ be an $|\Re|$-dimensional vector specifying the limited supply of each resource. Considering an acyclic, directed graph G, let $n=|V(G)|$ and $m=|A(G)|$, and order nodes topologically so that every arc $(i,j)\in A(G)$ satisfies $i<j$. Note that $V(\bullet)$ and $A(\bullet)$ denote the set of nodes and arcs of graph $\bullet$, respectively. Each arc (i,j) has an associated cost c_{ij} and a vector $u_{ij}=(u_{ij1},\cdots,u_{ij|\Re|})$, indicating the amount of each resource required to traverse the arc. Let path v_1-v_j denote a series of consecutive arcs from start node v_1 to node v_j (such a path may not be unique). The requirement of resource r (cost) on a path is the sum of the requirements of resource r (costs) associated with all arcs on that path. RCSP is to find the shortest path (i.e., the path with the least total arc cost) from start node v_1 to end node v_n with a total requirement of each type of resource $r\in\Re$ that observes a given limited supply T_r.

RCSP is often used as a sub-problem in branch-and-price (B&P) [1] and has been successful in solving well-known problems like crew scheduling [2,3], prescribing the content and timing of products upgrades [4], optimizing picking and placing operations on dual-head placement machines [5–7], and multi-commodity flow problems [8]. Many applications are modeled using an acyclic graph and/or multiple resource restrictions (e.g., [2–7] mentioned above, nurse scheduling [9], crew pairing [10], simultaneous scheduling of pilots and aircraft flights [11], and transfer line balancing [12]). A resource, for example, can be time, distance, capacity, money, workload, or reliability requirement.

Motivated by these applications, Zhu and Wilhelm [13] studied RCSP with one or more resource constraints on an acyclic graph in the context of column generation in which arc costs change at each iteration but resource constraints do not. Zhu and Wilhelm [13] proposed a three-stage approach (TSA), analyzed its worst-case complexity, and evaluated it computationally, comparing its performance with that of CPLEX as well as the state-of-the-art label-setting algorithm of Dumitrescu and Boland [14], showing that TSA is effective in repeatedly solving the dynamic RCSP (e.g., in column generation).

When RCSP is used as a sub-problem in column generation, arc costs (i.e., objective function coefficients) are updated using new (optimal) values of dual variables from the master problem but resource constraints remain the same at each column-generation iteration. Consequently, RCSP must be reoptimized with respect to these new arc costs, giving rise to the *dynamic property* of RCSP. We refer to the dynamic RCSP as the underlying graph and resource constraints are fixed, but arc costs are changed dynamically at each column-generation iteration. Even though label setting algorithms can apply effective dominance criteria to restrict the number of alternative paths they analyze, they suffer from the need to solve the entire problem from scratch each time

* Corresponding author. Fax: +1 865 9740588.
E-mail addresses: xzhu5@utk.edu (X. Zhu), wilhelm@tamu.edu (W.E. Wilhelm).

0305-0548/$ - see front matter © 2012 Published by Elsevier Ltd.
http://dx.doi.org/10.1016/j.cor.2012.07.007

Table 1
Acronyms.

B&B	Branch-and-bound	B&P	Branch-and-price
MDFA	Method for dealing with fixed arcs	OA	Optimizing algorithm used in stage 3 of TSA
RCSP	Resource-constrained SPP	ROA	Reoptimizing algorithm used in stage 3 of TSA
SPP	Shortest-path problem	TSA	Three-stage approach

arc costs change. In contrast, TSA deals only with resources in the first two stages, so they must be implemented only once in solving the overall problem, and the third stage, which is solved repetitively, is a shortest path problem in which arcs costs are changed at each iteration.

The current paper extends this line of research, investigating two issues with the goal of effectively solving the dynamic RCSP in a B&P context. First, when RCSP is used as a sub-problem in column generation, it is typical that costs are updated for a different subset of arcs on each column-generation iteration. We propose a reoptimizing algorithm (ROA) to generate the new solution by updating the last solution using revised (i.e., different) arc costs and incorporate ROA in TSA. Second, when RCSP is used as a sub-problem in B&P, some arcs in the graph may be fixed: forbidden or prescribed (i.e., associated decision variables fixed to 0 or 1, respectively, by the branching rule). We propose a method for dealing with fixed arcs (MDFA) and incorporate it in TSA.

The research objectives of this paper are (i) specialized algorithms (i.e., ROA and MDFA) for RCSP on an acyclic graph that can handle issues related to reoptimization as well as arcs that are fixed by branching within B&P; (ii) complexity analysis of these algorithms; (iii) computational evaluation; and (iv) implementation issues for solving the dynamic RCSP in B&P. These specialized algorithms are incorporated in TSA to make it suitable for solving the dynamic RCSP in B&P. Consequently, they can speed up B&P to solve problems (e.g., the scheduling problems in [2,3]) that involve RCSP as a sub-problem. It is worth mentioning that the goal of our computational tests is to evaluate the performances of ROA and MDFA imbed within TSA to solve the dynamic RCSP, rather than column generation or B&P itself. For reader convenience, Table 1 summarizes the acronyms used in this paper.

The remainder of this paper is structured as follows. Section 2 reviews related literature and Section 3 gives an overview of TSA and related propositions. Section 4 proposes ROA and presents a version of TSA that incorporates ROA. Section 5 proposes MDFA and presents a version of TSA that incorporates MDFA. Section 6 presents a set of tests designed to evaluate the effectiveness of ROA and MDFA. Finally, Section 7 gives conclusions and discusses implementation issues of these algorithms in B&P.

2. Literature review

RCSP is NP-hard [15], even if the graph is acyclic, only one resource constraint is involved, and all resource requirements and costs are positive [14]. RCSP on an acyclic graph is NP-hard in the ordinary sense and can be solved in pseudo-polynomial time [17]. The solution to RCSP is guaranteed to be elementary (no node is visited more than once) if arc costs are nonnegative and resource constraints have only upper bounds, or if the graph is acyclic as in this paper [16].

Because a number of important practical problems imbed RCSP, it has been studied rather extensively. Handler and Zang [15], Beasley and Christofides [16], and Mehlhorn and Ziegelmann [18] used methods that involve solving a relaxed problem using Lagrangian or linear relaxation. Others (e.g., [14,19–26]) used dynamic programming. These methods did not explicitly consider repeated solution of the dynamic RCSP as a sub-problem in column generation. In this context, the dynamic RCSP must be solved repeatedly, each time with an updated set of arc costs but the same set of resource constraints. To reoptimize at each column-generation iteration, prior algorithms must be employed starting from scratch. In contrast, several approaches exploit the column-generation context. Zhu and Wilhelm [13] proposed TSA for use in repeatedly solving the dynamic RCSP on an acyclic graph. Wilhelm et al. [4] reported a similar, but much less efficient, two-phase approach for a layered acyclic graph in which each arc is incident from a node in one level to another node in the next level. However, neither study specially investigated the issues of reoptimization and fixed arcs in the context of B&P. This paper proposes methods to deal with these two issues and specializes TSA, incorporating the proposed methods for repeatedly solving the dynamic RCSP. Section 3 gives a brief review of TSA so that this paper is self-contained.

In early work related to reoptimizing SPP, Goto and Sangiovanni-Vincentelli [27] investigated the problem of updating shortest paths from all nodes to a selected set of nodes for the case in which the cost on each of a subset of arcs is decreased. Their method, which is based on LU factorization of arc cost matrix, $\{c_{ij}\}$, requires a considerable amount of memory and is effective only if matrix $\{c_{ij}\}$ is sparse. Gallo [28] adapted Dijkstra's shortest-path algorithm [29] for reoptimizing a SPP in two cases: (i) a different node is selected to be the start node of SPP, and (ii) exactly one arc is assigned a new cost that is less than the previous value. Fujishige [30] proposed another Dijkstra-like method for the case in which each of a set of arcs incident to one node is assigned a new cost that is less than the previous value. Recently, Buriol et al. [31] proposed a technique that can reduce the sizes of heaps used by several reoptimization algorithms [32–34] for the case in which a single arc is assigned a new cost that is either smaller or larger than the previous value. They presented a comprehensive computational evaluation of their technique and provided a survey of research that dealt with the case in which the cost of a single arc is changed. These methods are very restrictive and apply only to special cases (e.g., exactly one arc is assigned a new cost, or each of a set of arcs incident to one node is assigned a new cost). Although they may be applied iteratively if several arc costs change, the resulting algorithms are computationally impractical when a number of arc costs are assigned new values.

Pallottino and Scutellà [35] proposed a methodology to reoptimize SPP on an arbitrary graph for the case in which the cost on each of a specified subset of arcs is either increased or decreased. They generalized previous work [28,30], devising a two-phase method. The dual phase sequentially reoptimizes arcs with increased costs. The primal phase dynamically decomposes the set of arcs with decreased costs into disjoint subsets and reoptimizes subsets sequentially. As a sub-problem in column-generation, RCSP must be reoptimized at each column-generation iteration with respect to new arc costs but subject to the same set of resource constraints. The optimal solution (i.e., a shortest path tree rooted at start node) from the last iteration is available. This paper presents ROA to reoptimize the dynamic RCSP on an acyclic graph if the costs of any subset of arcs are changed to values that are either smaller or larger than the previous ones.