Transportation Research Part B 46 (2012) 1043–1067

Contents lists available at SciVerse ScienceDirect

Transportation Research Part B

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

Parametric search and problem decomposition for approximating Pareto-optimal paths

Chi Xie [a,*], S. Travis Waller [b,1]

[a] Center for Transportation Research, Department of Civil, Architectural and Environmental Engineering, University of Texas at Austin, United States
[b] School of Civil and Environmental Engineering, University of New South Wales, Australia

ARTICLE INFO

Article history:
Received 22 October 2011
Received in revised form 14 March 2012
Accepted 14 March 2012

Keywords:
Pareto-optimal solution
Biobjective shortest path
Constrained shortest path
Parametric algorithm
Multiobjective problem decomposition

ABSTRACT

The multiobjective shortest path problem arises in many transportation and logistics applications, either as a stand-alone network routing problem or a subroutine of a more complex multiobjective network optimization problem. It has been addressed by different solution strategies, including labeling methods, ranking methods, constraint methods, and parametric methods. Increasing attention has been paid to parametric methods in recent years, partially because of its simple algorithmic logic and its flexibility of being used in different user-preference decision-making environments. The core idea of a parametric algorithm is scalarization, by which a multiobjective shortest path problem can be tackled by repeatedly solving a single-objective subproblem. However, existing parametric algorithms suffer two notorious deficiencies, which considerably limit its further applications: first, typical subroutines for the single-objective subproblem in general cannot capture nonextreme Pareto-optimal paths; second, parametric algorithms for biobjective problems cannot be directly extended to solving multiobjective problems. This paper provides some algorithmic improvements that can partially overcome these deficiencies. In particular, the contribution of this work is twofold: first, in the biobjective parametric solution framework, we propose an approximate label-setting algorithm for the parameterized, constrained single-objective subproblem, which is capable of identifying all extreme paths and a large percentage (i.e., 85–100%) of nonextreme paths; second, we suggest a general projection scheme that can decompose a multiobjective problem into a number of biobjective problems. The approximate parametric algorithm runs in polynomial time. The algorithmic design and solution performance of the algorithm for multiobjective shortest path problems are illustrated, and numerically evaluated and compared with a benchmark algorithm in terms of solution completeness and efficiency.

1. Introduction

Many optimal routing problems for transportation networks explicitly or implicitly contain multiple objectives, such as time, distance, cost, reliability, risk, and so on. On the one hand, these multiobjective routing problems occur on their own and have direct applications in network analysis; on the other hand, they often exist as algorithmic subroutines for solving more complicated multiobjective network optimization problems, such as minimum cost flow problem, multicommodity flow problem, and optimal network design problem in the form of multiple objectives. Potential applications of multiobjective optimal routing problems are numerous and still growing. It is impossible to give a complete list of these applications

* Corresponding author. Tel.: +1 512 232 3120; fax: +1 512 232 3153.
E-mail addresses: chi.xie@mail.utexas.edu (C. Xie), s.waller@unsw.edu.au (S. Travis Waller).
[1] Tel.: +61 (2) 9385 5721; fax: +61 (2) 9385 6139.

0191-2615/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.trb.2012.03.005

here. For an extensive review of multiobjective routing models and applications, interested readers are referred to Current and Min (1986), Current and Marsh (1993),White (1990), Boffey (1995), Ulungu and Teghem (1994), Ehrgott and Gandibleux (2000), and Raith and Ehrgott (2009).

In urban transportation networks, motorists often face route choices between, for example, minimizing travel time and minimizing monetary cost (e.g., toll, fare, and vehicle operating cost), or minimizing expected travel time and maximizing on-time arrival reliability. Given multiple objectives, routing decisions typically vary among different travelers, because of their different values of time that are mostly determined by their income levels and trip purposes, or their different preferences or perceptions on travel time uncertainty. To satisfy varying individual routing preferences, it is expected that an in-vehicle route guidance system can provide all possible, or at least a representative part of, attractive candidate paths to travelers, who then can make their own routing decisions from the candidate set. Finding such a path set poses a multiobjective optimal routing problem.

In the freight transportation field, an important and frequently referenced multiobjective routing problem arises from hazardous material transportation. Given the different concerns from the shipper, carrier and public sector, the route planner aim to find a set of paths for consideration, which represent different trade-offs between a few competing objectives: (1) minimizing travel distance (or travel time or transportation cost); (2) minimizing the size of exposed populations; (3) minimizing accident rate (for example, see Turnquist, 1987; Nozick et al., 1997; Chang et al., 2005; Ekrut et al., 2007). While the second and third objectives presented here could be reasonably combined to generate a performance measure named risk exposure, which is simply the product of population size and accidental rate, further combination seems less reasonable. Nevertheless, in either the triobjective or biobjective routing case, major changes with model and algorithmic complexity occur as the number of objectives increases from one to more, which poses a multiobjective shortest path problem.

The shortest path problem with more than one objective or attribute may be stated in the following way. Suppose a multi-attribute directed network $G = (V, E, N)$, where V is the vertex or node set, E is the edge or arc set, and N is the attribute set of edges or arcs. Every arc $(i,j) \in E$ is associated with a set of attributes specified by N and its attribute vector is $c_{ij} = (c_{1,ij}, c_{2,ij}, \ldots, c_{n,ij})$, where $|N| = n$. Given origin node r and destination node s, a shortest path problem with n objectives is to find all nondominated or noninferior or efficient paths between r and s in terms of the n path attributes, where it is assumed that all arc attributes are *nonnegative* and *additive* along paths, or in other words, this shortest path problem is of the *min–sum* type. Such an n-objective shortest path problem can be in general written as,

$$\min \quad \mathbf{z}(\mathbf{x}) = [z_k(\mathbf{x})]_{k=1,\ldots,n} = \left[\sum_{(i,j)\in E} c_{k,ij} x_{ij}\right]_{k=1,\ldots,n} \tag{1}$$

$$\text{s.t.} \quad \sum_{\{j:(i,j)\in E\}} x_{ij} - \sum_{\{j:(j,i)\in E\}} x_{ji} = \begin{cases} 1 & i = r \\ 0 & \forall i \in V - \{r,s\} \\ -1 & i = s \end{cases} \tag{2}$$

$$x_{ij} = 0 \quad \text{or} \quad 1 \qquad \forall (i,j) \in E \tag{3}$$

The functional form of the multiobjective shortest path problem is simply an extension of its single-objective counterpart. However, it should be noted that, different from the single-objective problem, the integrality constraint (3) is required in the multiobjective model; otherwise, optimal solutions may contain non-integral numbers, which fail to represent individual paths. Due to the imposition of such an integrality requirement, an n-objective shortest path problem is known as an NP-hard problem in its worst case, even if n is as small as 2 (see Garey and Johnson, 1979; Serafini, 1986).

The main concern of this paper is on the development of an approximate polynomial-time parametric procedure for biobjective shortest path problems, which is potentially capable of finding the full or near full set of Pareto-optimal paths. We also devise a simple decomposition procedure that can convert a multiobjective problem into a number of biobjective problems. In favor of this decomposition tool, a multiobjective shortest path problem can be indirectly solved by a biobjective algorithm. Specifically, we discuss these algorithmic developments in the remaining part of this paper as follows. In Section 2, we categorize and review previous multiobjective shortest path algorithms in the literature. Then the existing parametric solution framework and our new algorithmic advance are elaborated and illustrated in Section 3. Following this, in Section 4 we describe and prove a problem decomposition scheme for problems with three or more objectives. These methodological developments are then evaluated and compared with a benchmark algorithm through numerical experiments, whose results are discussed in Section 5. Finally, we summarize and conclude the paper in Section 6.

2. Previous work

In this section, we review solution methods that are designed to find a complete set of Pareto-optimal solutions for a multiobjective shortest path problem, under the assumption that the routing planner does not have the attribute or utility preference information a priori. In terms of algorithmic principles and operational characteristics, existing solution methods may be grouped into four categories, namely, labeling methods, ranking methods, constraint methods, and parametric methods. Since we only consider multiobjective routing problems with deterministic and time-invariant arc attributes, previous

studies on problems in stochastic and/or dynamic networks, for example, Nielsen et al. (2003), Opasanon and Miller-Hooks (2006), Meng et al. (2009), to just name a few, are excluded here.

Multiobjective path-finding methods using labeling techniques are simply a multidimensional extension of its single-objective counterpart, as based on Bellman's principle of optimality for dynamic programming (DP) (Bellman, 1957). Similar to single-objective methods, multiobjective labeling methods include two schemes: label-setting methods (Brown and Strauch, 1965; Hansen, 1979; Daellenbach and De Kluyver, 1980; Martins, 1984; Cox, 1984; Turnquist, 1987; Tung and Chew, 1988, 1992) and label-correcting methods (Vincke, 1974; Loui, 1983; Corley and Moon, 1985; Warburton, 1987; Brumbaugh-Smith and Shier, 1989; Skriver and Andersen, 2000; Guerriero and Musmanno, 2001; Sastry et al., 2003), based on how label sets are updated at nodes and how shortest path labels "converge" to the optimal set. In labeling methods, the key feature distinguishing a multiobjective shortest path problem from a single-objective one is the storage of multi-label vectors and the use of the vector dominance rule at each node in the dynamic updating process. Due to the nondominated nature, the number of the label sets at a node could be as high as $\sum_{k=0}^{|V|-2}(|V|-2)!/(|V|-2-k)!$ in the worst case, where $|V|$ is the total number of nodes in the network (Tung and Chew, 1992).

The ranking methods for multiobjective routing problems resort to a k-shortest path finding technique. In a biobjective case of Clímaco and Martins (1982), a sequence of k shortest paths is generated with the first objective, where the value of k is a specified upper bound on the number of paths a priori, or the k value is a variable increasing until the path carrying the minimum of the second objective appears in the sequence. A vector dominance check is then conducted through the k-path list to exclude all dominated paths. The method might enumerate all the possible (dominated and nondominated) paths where k becomes a very large number. In its worst case, an ultimate k value equal to $(|V|-1)!$ could occur, resulting in an exponential increase in the computational effort (Mote et al., 1991). When this method is applied to shortest path problems with more than two objectives, the computation cost becomes significantly more intractable, even for a network of moderate size.

Multiobjective shortest path problems are closely related to constrained shortest path problems. The third approach, which makes use of this structural relation, is to convert all but one of the objectives to upper-bound constraints and solve a set of resulting constrained single-objective shortest path problems. Lawler (1976) used this method to solve a biobjective problem. The algorithmic procedure is similar to that of the goal programming method (see Ignizio, 1976). Though the constraint method seems intuitively appealing, the constrained shortest path problem itself is not easy to solve (i.e., it is an integer programming (IP) problem). While this method might be reasonable for solving biobjective problems, it does not appear to be effective when the number of objectives increases (Boffey, 1995). The computational efficiency and solution quality may be greatly discounted when a large number of dominated solutions are generated while nondominated solutions are omitted during the process of forming constraints and solving constrained problems.

Parametric methods (or sandwich methods in the context of other multiobjective optimization problems) for dealing with multiobjective shortest path problems appear in the literature in two types. The first type uses a parameterized utility function combining all the objectives so as to convert a multiobjective problem into a series of single-objective shortest path problems with a range of parameter values (Robbins, 1983; Henig, 1985; Current et al., 1990; Coutinho-Rodrigues et al., 1999). Each parameter in the utility function serves as a weight for its corresponding objective; Pareto-optimal solutions are obtained sequentially through exhausting the parameter range and solving the corresponding parameterized problems. This solution strategy originates from Cohon's (1978) noninferior set estimation (NISE) method for multiobjective linear programming (MOLP) problems. The second type resorts to a linear relaxation of the original integer-restricting functional form of the multiobjective problem (White, 1982; Mote et al., 1991). Extreme nondominated solutions, which all belong to the extreme solution set of the relaxed linear program, can be readily identified by the pivot operation of a multiobjective simplex method. While both types of parametric methods resort to a linear relaxation to the original problem, the difference between them is that the former type of methods solves a set of single-objective LP problems while the latter directly deals with a multiobjective LP problem.

The computational efficiency of either type of parametric methods is embodied by their algorithmic design proposed for minimizing the use of the computationally intensive dominance check of the labeling methods and the inevitable redundant search of dominated paths of the ranking methods. A common algorithmic feature pertaining to the parametric methods is that Pareto-optimal solutions are generated individually as a sequence, each of which can be solved in general by some polynomial-time algorithm. Those "independent" individual solution searches may be conducted in a parallel manner, which further reduces the computational cost through implementing parallel processing techniques. Moreover, as needed, the order of generating individual solutions may be customized to give priority to those most "representative" solution points. This characteristic is very useful since in many cases it is not necessary to present all the Pareto-optimal solutions to the decision maker but a subset of key solutions supporting the Pareto-optimal profile. In an actual computational process, we may suspend the solution search when the number and range of collected Pareto-optimal solutions satisfy a certain threshold. Just due to this flexibility, a parametric method can be used in a variety of different objective-preference information environments, including a priori, a posteriori or interactive preference articulation cases.

However, the deficiency of existing parametric methods is also apparent: they can only find extreme Pareto-optimal solutions, which are those minimizing convex combinations of individual objectives. Unless all the objectives are commensurate to each other, the extreme solution subset cannot in general represent the whole Pareto-optimal set. The reason for this partial solution set problem is that use of a utility function or linear relaxation yields a more aggressive vector dominance condition, which in principle omits all nonextreme solutions. (This redundant solution dominance phenomenon is illustrated in

the next section.) Due to this reason, parametric methods are often used jointly with other methods for finding nonextreme Pareto-optimal solutions. In fact, such a mixed solution strategy has been suggested in a few algorithm implementations. For example, Current et al. (1990) used their NISE-like parametric algorithm with an auxiliary constrained shortest path algorithm; Coutinho-Rodrigues et al. (1999) suggested a combination of an NISE-like algorithm and a k-shortest path algorithm; Mote et al. (1991) developed a two-phase procedure that uses a multicriterion simplex method in the first phase and then a label-correcting method in the second phase; the two-phase algorithmic idea was also implemented in Raith and Ehrgott (2009) who suggested a hybrid solution strategy in which a parametric method is used in combination with a labeling or ranking method.

In addition, it must be noted that though these parametric methods have been successfully developed for biobjective shortest path problems, it does not seem straightforward to extend the algorithmic procedure to deal with problems with three or more objectives. The same deficiency exists in parametric methods for general multiobjective optimization methods (Rennen et al., 2011). Some researchers, e.g., Boffey (1995), suggested treating extra objectives (in addition to two) as side constraints or subsidiary objectives. Such a treatment, however, might ignore some attractive Pareto-optimal solutions to the original problem. Because of this difficulty, it is not surprising that all the parametric methods listed above have only been applied to biobjective problem cases. Exceptions are due to recent contributions by Przybylski et al. (2010a), and Özpeynirci and Köksalan (2010) who extended the parametric method to IP problems with three or more objectives, and by Rennen et al. (2011) who did a similar extension for general nonlinear programming (NLP) problems with multiple objectives.

The focus of this paper is given to some algorithmic improvements that can largely overcome the two inherent deficiencies pertaining to a generic parametric method when it is applied to finding Pareto-optimal paths. For the sake of completeness, some necessary definitions and principles related to multiobjective optimization problems are first stated in the following section.

3. Parametric algorithm for biobjective problems

This section begins with a few fundamental definitions for multiobjective optimization problems and then elaborates the proposed algorithmic framework and steps for biobjective shortest path problems.

Definition 1 (*Pareto-optimal condition*). The general definition of Pareto-optimality is closely related to the vector solution dominance. Given an n-objective shortest path problem, $\min \mathbf{z}(\mathbf{x}) = [z_k(\mathbf{x})], \forall k = 1, 2, \ldots, n$, where $\mathbf{x} = [x_{ij}], \forall (i,j) \in E$, satisfies constraints in (2) and (3), we say that solution $\mathbf{x}^a$ dominates solution $\mathbf{x}^b$ if and only if $\mathbf{z}(\mathbf{x}^a) \leqslant \mathbf{z}(\mathbf{x}^b)$ holds, i.e., $z_k(\mathbf{x}^a) \leqslant z_k(\mathbf{x}^b)$, $\forall k$, and at least one of the inequalities strictly holds, i.e., $z_k(\mathbf{x}^a) < z_k(\mathbf{x}^b)$, $\exists k$. Thus, an equivalent condition to the Pareto-optimality or nondominance of a feasible solution $\mathbf{x}^*$ to the multiobjective shortest path problem is that there does not exist any other feasible solution that dominates $\mathbf{x}^*$.

Definition 2 (*Extreme and nonextreme Pareto-optimal solutions*). The notion of extreme solutions plays an important role in understanding the algorithmic behavior of a parametric method. We say that a solution $\mathbf{x}$ to a multiobjective shortest path problem formulated as (1)–(3) is an extreme solution, if it cannot be represented as a strict convex combination of any two distinct feasible solutions in the feasible region F' of the *integer-relaxing problem*, where F' is defined by constraints (2). In other words, an extreme solution $\mathbf{x}$ is not equal to $\lambda_1\mathbf{x}_1 + \lambda_2\mathbf{x}_2$, given any $\mathbf{x}_1, \mathbf{x}_2 \in F\prime$, and $\lambda_1 + \lambda_2 = 1$, $0 \leqslant \lambda_1, \lambda_2 \leqslant 1$, if $\mathbf{x} \neq \mathbf{x}_1$ and $\mathbf{x} \neq \mathbf{x}_2$. Extreme and nonextreme Pareto-optimal solutions are those nondominated solutions satisfying and dissatisfying this extreme condition, respectively.

Definition 3 (*Full and partial Pareto-optimal objective sets*). A full Pareto-optimal or nondominated objective set to an n-objective shortest path problem is an n-dimensional objective vector that satisfies the conditions described in Definition 1. The solution corresponding to this full nondominated objective set is called an n-nondominated solution. A partial Pareto-optimal or nondominated objective set to the same problem is an m-dimensional objective vector that satisfies the Pareto-optimal condition in terms of its m objectives, where $2 \leqslant m < n$. Its corresponding solution is called an m-nondominated solution.

Suppose an arbitrary feasible solution to an n-objective shortest path problem. As we show later, if this solution is Pareto-optimal in terms of a partial objective set and does not have the exactly same objective values as any other solution in terms of the partial objective set, it is Pareto-optimal in terms of the full objective set; if the solution is Pareto-optimal in terms of the full objective set, it is Pareto-optimal in terms of at least one of its partial objective sets.

3.1. Parametric solution framework

The algorithmic motivation of a parametric algorithm resembles the use of Lagrangian multipliers in solving an optimization problem with side constraints. In a Lagrangian relaxation method, it is well known that the complexity reduction benefit of a complicated problem is attributed to relaxing some "hard" side constraints and supplementing them into the objective function as Lagrangian terms. By doing so, the original problem can then be tackled by repeatedly solving the

relaxed problem with updating Lagrangian multipliers. In a similar spirit, we may find the Pareto-optimal set through repeatedly solving a "relaxed" single-objective problem whose objective function is a combination of all original objectives with varying parameter values. The basic operation for such a parametric algorithm is to update the parameter set so as to find different Pareto-optimal solutions.

In its most frequently used form, a biobjective shortest path problem may be parameterized to be a single-objective problem via a convex combination:

$$\min \quad z_{\mathbf{w}}(\mathbf{x}) = \mathbf{w} \cdot \mathbf{z}(\mathbf{x}) = \sum_{l=1,2} w_l z_l(\mathbf{x}) = \sum_{(i,j)\in E}\sum_{l=1,2} w_l c_{l,ij} x_{ij} \tag{4}$$

$$\text{s.t.} \quad \sum_{\{j:(i,j)\in E\}} x_{ij} - \sum_{\{j:(j,i)\in E\}} x_{ji} = \begin{cases} 1 & i = r \\ 0 & \forall i \in V - \{r,s\} \\ -1 & i = s \end{cases} \tag{5}$$

$$x_{ij} \geqslant 0 \qquad \forall (i,j) \in E \tag{6}$$

where $\mathbf{w} = (w_1, w_2)$ is the parameter set and $\sum_{l=1,2} w_l = 1$ and $w_l \geqslant 0, \forall l$, hold. It can be seen that the parameterized problem is a standard shortest path problem, where the attribute of arc (i,j) is a convex combination of the original arc attribute set, i.e., $c_{\mathbf{w},ij} = \sum_{l=1,2} w_l c_{l,ij}$. Note that after this transformation, the integrality constraint (3) is no longer required for the parameterized problem; instead, a nonnegative constraint (6) $x_{ij} \geqslant 0, \forall (i,j) \in E$ is used.

A generic algorithmic procedure of searching the parameter space and finding Pareto-optimal solutions can be described and its validness proved as follows. The algorithm gets started with locating two initial Pareto-optimal solutions using two parameter sets: $\mathbf{w}_1 = (1-\varepsilon, \varepsilon)$ and $\mathbf{w}_2 = (\varepsilon, 1-\varepsilon)$, where ε is a sufficiently small number, i.e., $0 < \varepsilon \ll 1$. It is apparent that the values of $\mathbf{w}_1$ and $\mathbf{w}_2$ are so set as to obtain two extreme Pareto-optimal solutions that are optimal to the first and second objectives, respectively, over the whole feasible solution region. Use of a small ε value in these initial parameter sets avoids the risk of choosing a dominated solution when multiple optimal solutions to an initial relaxed problem are tied. Suppose that the two Pareto-optimal solutions obtained by solving the two initial parameterized problems at iteration 0 are $\mathbf{z}_1^{(0)} = \left(z_{1,1}^{(0)}, z_{1,2}^{(0)}\right)$ and $\mathbf{z}_2^{(0)} = \left(z_{2,1}^{(0)}, z_{2,2}^{(0)}\right)$. According to the Pareto-optimality condition, we then know that in a biobjective case, $z_{1,1}^{(0)} \leqslant z_{k,1} \leqslant z_{2,1}^{(0)}$ and $z_{1,2}^{(0)} \geqslant z_{k,2} \geqslant z_{2,2}^{(0)}$, where $\mathbf{z}_k = (z_{k,1}, z_{k,2})$ is one of subsequent Pareto-optimal solutions, as identified by the parametric algorithm at the kth iteration.

After the initial phase, the major operations of the algorithm includes generating new parameter sets and searching for new Pareto-optimal solutions in terms of generated parameter sets. The whole procedure can be in some sense depicted as an iterative divide-and-conquer process, which either locates a Pareto-optimal solution in each parameter range confined by the two previous Pareto-optimal parameters that are used to generate the current parameter set, or concludes that no Pareto-optimal solution is found in this range.

At each iteration, the parameter set is computed based on two neighboring Pareto-optimal solutions generated from some previous iterations. The most widely used method of generating parameters is the perpendicular method, which results in a parameter vector perpendicular to the line going through the two Pareto-optimal solution points. In addition to those biobjective problem instances listed in the last section, the perpendicular method has also been used in parametric methods for other biobjective optimization problems, such as Cohon (1978) for general biobjective LP problems, Aneja and Nair (1979) for a biobjective transportation problem (a special LP problem), and Fruhwirth et al. (1989) for a biobjective minimum cost flow problem (a special LP problem). Specifically, given two Pareto-optimal solutions $\mathbf{x}_1$ and $\mathbf{x}_2$, and their corresponding objective vectors $\mathbf{z}_1 = (z_{1,1}, z_{1,2})$ and $\mathbf{z}_2 = (z_{2,1}, z_{2,2})$, the perpendicular method generates a new parameter set using the following linear system,

$$\begin{bmatrix} z_{1,1} - z_{2,1} & z_{1,2} - z_{2,2} \\ z_{2,1} - z_{1,1} & z_{2,2} - z_{1,2} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} \frac{a_1}{a_1+a_2} \\ \frac{a_2}{a_1+a_2} \end{bmatrix} \tag{7}$$

where $a_1 = z_{2,2} - z_{1,2}$ and $a_2 = z_{1,1} - z_{2,1}$. It is readily seen that as long as a_1 and a_2 are simultaneously positive or negative, the linear system has a solution for $\mathbf{w}$. In fact, this is a sufficient and necessary condition for two different solutions $\mathbf{x}_1$ and $\mathbf{x}_2$ being Pareto-optimal. In other words, if $\mathbf{x}_1$ and $\mathbf{x}_2$ are two different Pareto-optimal solutions, there must exist a parameter set $\mathbf{w}$ satisfying $\sum_{l=1,2} w_l = 1$ and $w_{l=1,2} \geqslant 0$.

The core process of this generic parametric procedure for biobjective shortest path problems includes the perpendicular method of generating the parameter set and a dynamic programming method solving the parameterized single-objective problem. If the parameterized problem is defined as an ordinary shortest path problem without any extra constraint, the optimal solution of this parameterized problem is and is only an extreme Pareto-optimal solution. This extreme Pareto-optimal solution is either a new Pareto-optimal solution or one of the two Pareto-optimal solutions that are used to generate the parameter set. In the latter case, we do not get any new Pareto-optimal solution corresponding to the parameter set at the current iteration and accordingly claim that there does not exist any extreme Pareto-optimal solution in the solution space confined by the two parameter-generating Pareto-optimal solutions.

The following conclusion guarantees that solving a parameterized shortest path problem in each iteration generates an extreme Pareto-optimal solution $\mathbf{x}^*$ satisfying the following condition:

$$\begin{cases} \min\{z_{1,1}, z_{2,1}\} < z_{*,1} < \max\{z_{1,1}, z_{2,1}\} \\ \min\{z_{1,2}, z_{2,2}\} < z_{*,2} < \max\{z_{1,2}, z_{2,2}\} \end{cases} \tag{8}$$

if such a solution exists in the above objective region, where $\mathbf{z}^* = (z_{*,1}, z_{*,2})$ is the objective vector of extreme Pareto-optimal solution $\mathbf{x}^*$.

Lemma 1. *At each iteration of the parametric method, given two extreme Pareto-optimal solutions, $\mathbf{x}_1$ and $\mathbf{x}_2$, and their corresponding objective vector values, $\mathbf{z}_1 = (z_{1,1}, z_{1,2})$ and $\mathbf{z}_2 = (z_{2,1}, z_{2,2})$, where $\mathbf{z}_1 \neq \mathbf{z}_2$, the optimal solution $\mathbf{x}^*$ of the parameterized problem with its parameter set generated by the perpendicular method is an extreme Pareto-optimal solution with its objective vector $\mathbf{z}^* = (z_{*,1}, z_{*,2})$ satisfying (8).*

Proof. Solving a parameterized problem is a repeated process of the parametric method. First, at iteration 0, in general we have two extreme Pareto-optimal solutions, which are obtained by optimizing each objective separately. These two Pareto-optimal solutions are used to generate the parameter set at iteration 1. The two Pareto-optimal solutions give the lower bound and upper bound of each objective in the whole Pareto-optimal set. It is apparent that all other Pareto-optimal solutions satisfy the condition in (8), including the Pareto-optimal solution generated at iteration 1. Moreover, this generated Pareto-optimal solution must be an extreme solution.

Suppose that at iteration k we have two extreme Pareto-optimal solutions at hand whose objective function values are different. And these two solutions are used to generate a new parameter set. Solving the parameterized shortest path problem with the new parameter set results in a new extreme Pareto-optimal solution, which must satisfy the condition in (8); otherwise, at least one of the two parameter-generating Pareto-optimal solutions is not an extreme solution, which contradicts the supposition. □

The lemma we prove above provides a necessary condition for the parameterized shortest path problem to be optimally solved by a dynamic programming procedure and guarantees the "convergence" of the complete extreme Pareto-optimal solution set. The sufficient and necessary conditions for the correctness of the generic parametric procedure are further validated by the following conclusion.

Lemma 2. *Given two Pareto-optimal solutions, $\mathbf{x}_1$ and $\mathbf{x}_2$, and their corresponding objective vectors, $\mathbf{z}_1 = (z_{1,1}, z_{1,2})$ and $\mathbf{z}_2 = (z_{2,1}, z_{2,2})$, we intend to construct a parameterized problem $\min z_{\mathbf{w}}(\mathbf{x})$ for identifying an extreme Pareto-optimal solution $\mathbf{x}^*$ to the original biobjective shortest path problem subject to the objective constraints in (8). Sufficient and necessary conditions for a dynamic programming algorithm to solve this problem are: (1) the objective function $z_{\mathbf{w}}(\mathbf{x})$ is a chained function; (2) $z_{\mathbf{w}}(\mathbf{x}) \leqslant z_{\mathbf{w}}(\mathbf{x}_1)$ and $z_{\mathbf{w}}(\mathbf{x}) \leqslant z_{\mathbf{w}}(\mathbf{x}_2)$ hold if $0 \leqslant z_{*,1} < \lambda_1 z_{1,1} + \lambda_2 z_{2,1}$ and $0 \leqslant z_{*,2} < \lambda_1 z_{1,2} + \lambda_2 z_{2,2}$, where there exists λ_1 and λ_2 satisfying $\lambda_1 + \lambda_2 = 1$ and λ_1, $\lambda_2 > 0$.*

Proof. This lemma may be regarded as a special case of a more general theorem regarding the sufficient and necessary conditions for a discrete optimization problem to be solved by dynamic programming (see Bonzon, 1970 for Theorem 4). This theorem claims the sufficient and necessary conditions for a discrete deterministic optimization problem to be solved by dynamic programming are (1) the graph of constraints must be a chained graph and (2) the objective function must be a chained function.

The sufficiency of the theorem is apparent. Specifically, the first condition justifies the use of dynamic programming for the parameterized problem (i.e., a standard shortest path problem); the second condition guarantees that the optimal solution to the parameterized problem, if obtained, is a solution to the biobjective shortest path problem satisfying the extreme Pareto-optimality defined in Definition 2.

Now let us suppose either condition 1 or 2 does not hold. If the objective function is not a chained function, the solution derived by a dynamic programming procedure may not be optimal to the parameterized problem and hence not be Pareto-optimal to the biobjective problem. If the second condition is not satisfied, the derived solution may be one of the given Pareto-optimal solutions that confine the range of the solution. So the necessity is proved. □

Apparently, the parameterized problem defined in (4)–(6) satisfies both conditions stated in Lemma 2. First, the objective function of this parameterized shortest path problem is a chained function along any path in the network. Second, it is clear that the objective function satisfies $\sum_{l=1,2} w_l z_l(\mathbf{x}^*) \leqslant \sum_{l=1,2} w_l z_l(\mathbf{x}_1) = \sum_{l=1,2} w_l z_l(\mathbf{x}_2)$, given that $\mathbf{w} = (w_1, w_2)$ is determined by (7), where $\mathbf{x}^*$ is the optimal solution of the parameterized problem.

Given a parameterized problem defined in (4)–(6), the generic parametric procedure, however, will return one of the given Pareto-optimal solutions used for generating the parameter set as the optimal solution, when no Pareto-optimal solution exists in the range confined by this parameter set (refer to (7)), or existing Pareto-optimal solutions are nonextreme in the range. The latter case means the arising of the redundant solution dominance problem, as we mentioned before. In fact, this undesirable solution dominance excludes all nonextreme solutions from the Pareto-optimal set. The underlying reason is from the optimality condition of the parameterized problem. It is noted that the parameterized problem is a (single-objective

or multiobjective) LP problem instead of an IP problem. Its optimal solution, that is, a Pareto-optimal solution to the original problem, can only be an extreme solution. Therefore, a parametric method whose single-objective subroutine is an ordinary LP or DP solution technique can never identify a nonextreme Pareto-optimal solution. As an alternative representation, this phenomenon may be more intuitively illustrated by an example in the objective space shown in Fig. 1.

Given two Pareto-optimal objective points, P_1 and P_2, the objective space of a biobjective problem may be partitioned into several regions. We are concerned about the following regions:

$$B_1 = \{(z_1, z_2) | z_1 \geqslant z_{1,1}, z_2 \geqslant z_{1,2}, (z_1, z_2) \neq (z_{1,1}, z_{1,2})\}$$
$$D_1 = \{(z_1, z_2) | z_1 \leqslant z_{1,1}, z_2 \leqslant z_{1,2}, (z_1, z_2) \neq (z_{1,1}, z_{1,2})\}$$
$$B_2 = \{(z_1, z_2) | z_1 \geqslant z_{2,1}, z_2 \geqslant z_{2,2}, (z_1, z_2) \neq (z_{2,1}, z_{2,2})\}$$
$$D_2 = \{(z_1, z_2) | z_1 \leqslant z_{2,1}, z_2 \leqslant z_{2,2}, (z_1, z_2) \neq (z_{2,1}, z_{2,2})\}$$
$$E = \{(z_1, z_2) | z_{2,1} < z_1 < z_{1,1}, z_{1,2} < z_2 < z_{2,2}, w_1 z_1 + w_2 z_2 < w_1 z_{1,1} + w_2 z_{1,2}\}$$
$$N = \{(z_1, z_2) | z_{2,1} < z_1 < z_{1,1}, z_{1,2} < z_2 < z_{2,2}, w_1 z_1 + w_2 z_2 \geqslant w_1 z_{2,1} + w_2 z_{2,2}\}$$

where w_1 and w_2 are defined in (7) and $w_1 z_{1,1} + w_2 z_{1,2} = w_1 z_{2,1} + w_2 z_{2,2}$.

Region B_1 is a region in which all points dominated by P_1, while all points in region D_1 dominate P_1. Similarly, P_2 dominates all points in region B_2 and all points in region D_2 dominate P_2. Note that the overlapped area of B_1 and B_2 covers those points dominated by both P_1 and P_2, while all points in the overlapped area of D_1 and D_2 dominate both P_1 and P_2. By definition, region E contains all possible extreme nondominated solution points to P_1 and P_2 (hereafter we term it an extreme objective region); region N is the nonextreme nondominated area to P_1 and P_2 (hereafter we term it a nonextreme objective region). It is readily known that given that P_1 and P_2 are two Pareto-optimal points, it is guaranteed that regions D_1 and D_2 do not contain any solution point. To this end, the task of the subroutine of a parametric method is to check whether there exists a solution point in region $E \cup N = \{(z_1, z_2) | z_{2,1} < z_1 < z_{1,1}, z_{1,2} < z_2 < z_{2,2}\}$, the Pareto-optimal region confined by P_1 and P_2. By using the generic parametric method described above, we can only find a Pareto-optimal solution with its objective point in region E, due to the convex combination form of the objective function of the parameterized problem, which implicitly imposes an extra constraint $w_1 z_1 + w_2 z_2 \leqslant w_1 z_{1,1} + w_2 z_{1,2} = w_1 z_{2,1} + w_2 z_{2,2}$ to the parameterized problem. If there exists no extreme nondominated objective point in region E, the subroutine procedure returns P_1 or P_2 (or another point on the line connecting P_1 or P_2).

To overcome the theoretical deficiency described above, we must reformulate the parameterized problem and eliminate the algorithmic limit pertaining to an LP problem that an optimal solution is always at its convex point or edge. Our focus is given to the design of a subroutine model and algorithm that are capable of finding both extreme and nonextreme Pareto-optimal solutions. In the meantime, we still want to take advantage of the algorithmic efficiency of dynamic programming processes so as to maintain the computational complexity at the polynomial-time level. Given these considerations, we need to overcome two algorithmic difficulties in designing a new parametric method: (1) the dynamic programming procedure must have the capability of searching the nonextreme objective region while avoiding to capture the two parameter-gener-

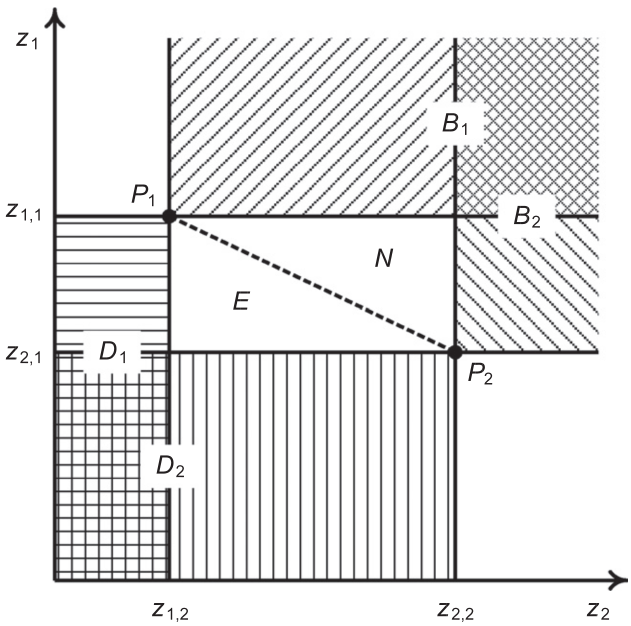


Fig. 1. Biobjective spaces partitioned by two Pareto-optimal solution points.

ating Parent-optimal solutions and (2) the dynamic programming procedure must ensure that the optimal solution satisfies the objective constraints stated in (8).

An approximate solution strategy is proposed below. It adds extra constraints to form a constrained shortest path problem. Its procedure is designed to explore the feasible objective spaces of both the extreme and nonextreme solutions (i.e., regions *E* and *N* in Fig. 1, as an example), which are confined by the two parameter-generating Pareto-optimal solutions, while excluding the possibility that either of the two Pareto-optimal solutions becomes the optimal solution of the parameterized problem.

3.2. Constrained shortest path approach

The constrained shortest path approach simply adds an extra constraint for each objective of the biobjective problem into the parameterized shortest path problem defined in (4)–(6):

$$\min \quad z_{\mathbf{w}}(\mathbf{x}) = \mathbf{wz}(\mathbf{x}) = \sum_{l=1,2} w_l z_l(\mathbf{x}) = \sum_{(i,j)\in E}\sum_{l=1,2} w_l c_{l,ij} x_{ij} \tag{9}$$

$$\text{s.t.} \quad \sum_{\{j:(i,j)\in E\}} x_{ij} - \sum_{\{j:(j,i)\in E\}} x_{ji} = \begin{cases} 1 & i = r \\ 0 & \forall i \in V - \{r,s\} \\ -1 & i = s \end{cases} \tag{10}$$

$$\sum_{\{j:(i,j)\in E\}} c_{l,ij} x_{ij} < \max(z_{1,l}, z_{2,l}) \qquad \forall l = 1,2 \tag{11}$$

$$x_{ij} = 0 \text{ or } \quad 1 \qquad \forall (ii,j) \in EE \tag{12}$$

Thus, the parameterized problem is a doubly-constrained shortest path problem, in which the upper bound for each objective is the larger one of the objective function values of the two parameter-generating Pareto-optimal solutions. Note that given the added constraints, the integrality constraint (12) must be included in the constrained shortest path problem, which introduces combinatorial complexity into the problem. In accordance with this variation of the problem formulation, the generic parametric algorithm turns to be a constrained parametric algorithm, in which the major modification is that the parameterized problem is a doubly-constrained shortest path problem. The following conclusion guarantees the correctness of the constrained parametric method for the biobjective shortest path algorithm.

Theorem 1. *Given two Pareto-optimal solutions, $\mathbf{x}_1$ and $\mathbf{x}_2$, and their corresponding objective values, z_1 and z_2, we intend to construct a parameterized problem $\min z_{\mathbf{w}}(\mathbf{x})$ in (9)–(12) for identifying a Pareto-optimal solution to the biobjective shortest path problem. The optimal solution to the parameterized, doubly-constrained shortest path problem, if exists, is a Pareto-optimal solution to the biobjective shortest path problem. The collection of optimal solutions of all such parameterized problems is the complete set of extreme and nonextreme Pareto-optimal solutions of the biobjective problem.*

Proof. It is clear that the objective function of the optimal solution to the parameterized, doubly-constrained shortest path problem satisfies the condition in (8). Since it is the optimal solution, no other feasible solution of the parameterized problem has a smaller objective function value, nor does other feasible solution dominate it (in the biobjective space). Therefore, the optimal solution to the parameterized problem is a Pareto-optimal solution to the biobjective shortest path problem.

The optimal solution of the parameterized, doubly-constrained shortest path problem can be either an extreme or nonextreme solution. Suppose that there exists a Pareto-optimal solution $\mathbf{x}^*$ of the biobjective shortest path problem not included in the Pareto-optimal set identified by the constrained parametric method. We can sort all the Pareto-optimal solutions in the increasing order in terms of one of the objectives, which results in a Pareto-optimal solution sequence that has the decreasing order in terms of the other objective (see Brumbaugh-Smith and Shier, 1989). Then we choose a pair of consecutive Pareto-optimal solutions, $\mathbf{x}_1$ and $\mathbf{x}_2$, whose objective values $\mathbf{z}_1$ and $\mathbf{z}_2$ satisfy $z_{1,1} < z_{*,1} < z_{2,1}$ and $z_{1,2} > z_{*,2} > z_{2,2}$. It is apparent that such a Pareto-optimal solution $\mathbf{x}^*$ can be obtained by solving a parameterized, doubly-constrained shortest path problem with its parameters determined by $\mathbf{x}_1$ and $\mathbf{x}_2$ using the perpendicular method (see (7)). This contradicts our hypothesis. Therefore, solving parameterized problems with all possible parameters will result in a complete set of Pareto-optimal solutions of the biobjective shortest path problem. □

Such a constrained shortest path problem, however, is an NP-hard problem due to the arising combinatorial complexity. A variety of exact and approximate solution algorithms have been developed for the constrained problem, including dynamic programming method (Aneja et al., 1983; Desrochers and Soumis, 1988; Desrosiers et al., 1995; Jaumard et al., 1996; Dumitrescu and Boland, 2001, 2003), Lagrangian relaxation method (Handler and Zang, 1980; Jaffe, 1984; Ribeiro and Minoux, 1985; Beasley and Christofides, 1989; Xiao et al., 2005; Carlyle et al., 2008), distributed method (Reeves and Salama, 2000; Sun and Langendörfer, 1998), and ε-approximation method (Hassin, 1992; Lorenz and Raz, 2001), among others. Some

of the algorithms listed here may perform with a pseudo-polynomial complexity for exact solutions, such as Hassin (1992), Jaffe (1984), Carlyle et al. (2008), Desrochers and Soumis (1988), and Dumitrescu and Boland (2003). In particular, the label-setting algorithm with preprocessing designed by Dumitrescu and Boland (2003) is believed as the most efficient procedure and shows consistently superior computation performance in extensive computation tests.

To bound the computational complexity to the polynomial-time level, we propose an approximate algorithm that can efficiently find an optimal or near-optimal solution through a dynamic programming process. This procedure has an analogous algorithmic structure to the label-setting algorithm with preprocessing presented by Aneja et al. (1983) and Dumitrescu and Boland (2003) for the singly-constrained shortest path problem. In fact, our approximate algorithm is an adapted and approximate version of the label-setting algorithm with preprocessing for doubly-constrained cases, which keeps all applicable polynomial-time steps while removing the exponentially complex dominance check step in the labeling process. This approximate procedure involves nothing more than repeatedly executing the label-setting algorithm for the standard shortest path problem and a polynomial number of simple arithmetic comparisons. The remaining text in this section briefly describes the major algorithmic mechanisms. For discussion convenience, the key notation used in the algorithm is contained in Table 1.

The approximate procedure can be described as two major algorithmic phases: *preprocessing* and *labeling* (see Fig. 2). The preprocessing phase is an *exact* procedure, which functions at searching for de facto unconstrained optimal solutions, examining the problem's solution feasibility, reducing the effective network size, and lowering the upper bound of the objective function. By exact, we mean that either of the following two solution conditions is exactly correct: first, the optimal solution identified by the preprocessing procedure for the constrained shortest path problem; second, the conclusion drawn by the preprocessing procedure that no feasible solution exists. Though the result of the preprocessing phase may include other cases beyond these two conditions, the preprocessing phase does not introduce any solution suboptimality. The operations of the preprocessing phase are arranged in three major steps in a loop. Specially, step 1 (including steps 1.1–1.3) explores the possibility of identifying the optimal solution via origin-based and destination-based shortest path searches in terms of the parameterized objective; step 2 (including steps 2.1–2.3) explores the possibility of identifying the problem infeasibility, updates the upper bound of the objective function, and provides useful constraint information for the labeling phase via origin-based and destination-based shortest path searches in terms of each of the original objectives; step 3 (including steps 3.1–3.2) explores the possibility of removing redundant nodes and arcs in the network that are not used by optimal paths. The efficacy of the preprocessing phase is due to the outcome that it not only significantly reduces the network size and limits the upper bound of the objective function, but also often identifies the optimal solution without resorting to the second phase. An extensive evaluation of the computational performance of the preprocessing phase for constrained shortest path problems can be referred to in Dumitrescu and Boland (2003).

The second phase is an *approximate* single-objective path search procedure subject to two additional solution feasibility checks. This approximate procedure maintains three labels for each node j obtained by searching for a constrained shortest path from the origin node to this node: one for the parameterized objective, $\bar{d}(j) = w_1\bar{c}_1(j) + w_2\bar{c}_2(j)$, and other two for the two original objectives, $\bar{c}_1(j)$ and $\bar{c}_2(j)$, respectively. The purpose of using the other two labels is to guarantee that any obtained solution satisfies the objective constraints (i.e., constraint (17)). This is simply realized by a feasibility check during the dynamic scanning and updating process, which is conducted for every node j when a new temporary label set for this node is formed through an upstream node i; it simply checks whether, for each objective l, the temporary label formed by a path from origin node r to this node j via an upstream node i, $\bar{c}_l(i,\overline{\Delta}(i)) + c_{l,ij}$, plus the permanent label formed in the preprocessing phase by the predetermined shortest path from destination node s to node j in terms of objective l, $c'_l(j)$, sat-

Table 1
Notation used in the constrained parametric algorithm.

Notation	Definition
Phase 1	
$\Gamma_l(i)$	Shortest path from origin node r to node i in terms of attribute l
$\Gamma'_l(i)$	Shortest path from destination node s to node i in terms of attribute l
$\Delta(i)$	Shortest path from origin node r to node i in terms of the combined attribute
$\Delta'(i)$	Shortest path from destination node s to node i in terms of the combined attribute
$c_l(i)$	Node label of attribute l generated by the shortest path from origin node r to node i in terms of attribute l
$c'_l(i)$	Node label of attribute l generated by the shortest path from destination node s to node i in terms of attribute l
$d(i)$	Node label the combined attribute generated by the shortest path from origin node r to node i in terms of the combined attribute
$d'(i)$	Node label of the combined attribute generated by the shortest path from destination node s to node i in terms of the combined attribute
$c_l(i,\Delta(i))$	Node label of attribute l generated by the shortest path from origin node r to node i in terms of the combined attribute
$c'_l(i,\Delta'(i))$	Node label of attribute l generated by the shortest path from destination node s to node i in terms of the combined attribute
$d(i,\Gamma_l(i))$	Node label of the combined attribute generated by the shortest path from origin node r to node i in terms of attribute l
Phase 2	
$\overline{\Delta}(i)$	Constrained shortest path from origin node r to node i in terms of the combined attribute
$\bar{c}_l(i,\overline{\Delta}(i))$	Node label of attribute l generated by the constrained shortest path from origin node r to node i in terms of the combined attribute
$\bar{d}(i)$	Node label of the combined attribute generated by the constrained shortest path from origin node r to node i in terms of the combined attribute
$A(i)$	Set of arcs emanating from node i

algorithm constrained parametric method for biobjective shortest path problems
begin
Step 0 **for** each $l = 1, 2$ **do**
solve single-objective shortest path problems in terms of arc attribute $c_{l,ij}$ from origin node r to all other nodes and from destination node s to all other nodes, respectively, by a label-setting algorithm, to get origin-based and destination-based shortest path trees $\Gamma_l(i)$ and $\Gamma'_l(i)$ and shortest path labels $c_l(i)$ and $c'_l(i)$, $\forall i$;
end
set $\mathbf{w}^{(0,p)}$, $p = 1, 2$
for each $\mathbf{w}^{(0,p)}$ **do**
solve a single-objective shortest path problem in terms of arc attribute $w_l^p c_{l,ij} x_{ij}$ by a label-setting algorithm, to get optimal solution $\mathbf{x}^{(0,p)}$ and its attribute set $\mathbf{z}_p^{(0)}$;
end
set $\mathbf{Q} := \left\{\left(\mathbf{z}_1^{(0)}, \mathbf{z}_2^{(0)}\right)\right\}$ and $\mathbf{R} := \emptyset$;
set $k := 1$;
while $|\mathbf{Q}| > 0$ and $k < k_{\max}$ **do**
set $Q := |\mathbf{Q}|$;
for $q := 1$ to Q **do**
Step 1 generate a two-dimensional parameter set $\mathbf{w}^{(k,q)}$ by the perpendicular method;
Step 2 solve a doubly-constrained single-objective shortest path problem by the subroutine (see Figure 3(b));
Step 3 **if** an efficient solution $\mathbf{x}^*$ is found **then**
set $\mathbf{Q} := \mathbf{Q} \backslash \left\{\left(\mathbf{z}_1^{(k,q)}, \mathbf{z}_2^{(k,q)}\right)\right\}$,
$\mathbf{Q} := \mathbf{Q} \cup \left\{\left(\mathbf{z}^*, \mathbf{z}_2^{(k,q)}\right), \left(\mathbf{z}_1^{(k,q)}, \mathbf{z}^*\right)\right\}$, and
$\mathbf{R} := \mathbf{R} \cup \left\{\left(\mathbf{z}_1^{(k,q)}, \mathbf{z}_2^{(k,q)}\right)\right\}$;
otherwise
set $\mathbf{Q} := \mathbf{Q} \backslash \left\{\left(\mathbf{z}_1^{(k,q)}, \mathbf{z}_2^{(k,q)}\right)\right\}$ and $\mathbf{R} := \mathbf{R} \cup \left\{\left(\mathbf{z}_1^{(k,q)}, \mathbf{z}_2^{(k,q)}\right)\right\}$;
end
end
set $k := k + 1$;
end
end

(a) Doubly-constrained parametric algorithm

Fig. 2. Constrained parametric algorithm and its subroutine for biobjective shortest path problems.

isfies the corresponding objective constraint, i.e., $\bar{c}_l(i, \overline{\Delta}(i)) + c_{l,ij} + c'_l(j) < \max(z_{1,l}, z_{2,l})$. After the feasibility check, another check is then conducted to examine whether the upper bound of the objective function is greater than the sum of the temporary label of node j, $\bar{d}(i) + \sum_l w_l^q c_{l,ij}$, and the permanent label formed in the preprocessing phase by the predetermined shortest path from destination node s to node j in terms of the parameterized objective, $d'(j)$, i.e., $\bar{d}(i) + \sum_l w_l^q d_l(j) + d'(j) < U$. If both the feasibility check and the upper bound check are passed, the temporary label for the parameterized objective, $\bar{d}(i) + \sum_l w_l^q c_{l,ij}$, is then compared to the existing parameterized label at this node to determine if an update for this label is required; otherwise, the temporary label is discarded. In the case that such an update is warranted, the whole set of labels, including the parameterized-objective label and the two single-objective labels, are updated with the new temporary label set.

Fig. 2 depicts the complete algorithmic steps of the constrained parametric algorithm (Fig. 2a) and its subroutine—the two-phase approximate label-setting algorithm (Fig. 2b)—for the doubly-constrained shortest path problem. As we commented earlier, this constrained parametric algorithm is capable of potentially finding both extreme and nonextreme Pareto-optimal solutions, due to the imposition of the objective constraints as well as the incorporation of this set of constraints in the exact preprocessing phase and the approximate labeling phase. The remaining concern about this algorithm is its solution efficiency and quality (or completeness).

It is clear that the major change of the constrained parametric method compared to the generic parametric method is that in each iteration a doubly-constrained shortest path problem is solved instead of a simple standard shortest path problem.

algorithm doubly-constrained single-objective shortest path subroutine
phase 1 preprocessing
begin

Step 0 set $U = d_{\max}(|V| - 1) + 1$, where $d_{\max} = \max_{(i,j)\in E} \sum_l w_l^p c_{l,ij}$;
while $G \subset G_0$ or $U < U_0$ **do**
set $G_0 = G$ and $U_0 = U$;
Step 1.1 solve the single-objective shortest path problem in terms of arc attribute $\sum_l w_l^p c_{l,ij}$ from origin node r to all other nodes by a label-setting algorithm to get origin-based shortest path tree $\Delta_l(i)$ and shortest path labels $d_l(i)$, $\forall i$;
Step 1.2 **if** $c_l(s, \Delta(s)) \le \max\left(z_{1,l}^{(k,q)}, z_{2,l}^{(k,q)}\right)$, $\forall l = 1, 2$ and $\Delta(s)$ is not identical to any one of the efficient paths corresponding to $\mathbf{z}_1^{(k,q)}$ and $\mathbf{z}_2^{(k,q)}$ **then**
conclude $\mathbf{x}(\Delta(s))$ is an efficient solution and return it to the main routine as $\mathbf{x}^*$;
end
Step 1.3 **if** $d(s) \ge U$ **then**
conclude the solution corresponding to U is an efficient solution and return it to the main routine as $\mathbf{x}^*$;
end
Step 1.4 solve the single-objective shortest path problems in terms of arc attribute $\sum_l w_l^p c_{l,ij}$ from destination node s to all other nodes by a label-setting algorithm to get destination-based shortest path tree $\Delta'_l(i)$ and shortest path labels $d'_l(i)$, $\forall i$;
for each $l = 1, 2$ **do**
Step 2.1 solve the single-objective shortest path problem in terms of arc attribute $c_{l,ij}$ from origin node r to all other nodes by a label-setting algorithm to get origin-based shortest path tree $\Gamma_l(i)$ and shortest path labels $c_l(i)$, $\forall i$;
Step 2.2 **if** $c_l(s) \ge \max\left(z_{1,l}^{(k,q)}, z_{2,l}^{(k,q)}\right)$ **then**
conclude no efficient solution is found and return to the main routine;
end
Step 2.3 **if** $\min\left(d(s, \Gamma_1(s)), d(s, \Gamma_2(s))\right) < U$ and the solution corresponding to $\min\left(d(s, \Gamma_1(s)), d(s, \Gamma_2(s))\right)$ is not identical to any one of the efficient paths corresponding to $\mathbf{z}_1^{(k,q)}$ and $\mathbf{z}_2^{(k,q)}$ **then**
set $U = \min\left(d(s, \Gamma_1(s)), d(s, \Gamma_2(s))\right)$ and record the corresponding solution;
end
Step 2.4 solve the single-objective shortest path problem in terms of arc attribute $c_{l,ij}$ from destination node s to all other nodes by a label-setting algorithm to get destination-based shortest path trees $\Gamma'_l(i)$ and shortest path labels $c'_l(i)$, $\forall i$;
end
Step 3.1 **for** each $i \in V\backslash\{r, s\}$ **do**
if $c_l(i) + c'_l(i) \ge \max\left(z_{1,l}^{(k,q)}, z_{2,l}^{(k,q)}\right)$, $\exists l = 1, 2$ **then**
remove node i and all arcs incident to it from network G;
otherwise
if $d(i) + d'(i) \ge U$ **then**
remove node i and all arcs incident to it from network G;
end
end
end

Fig. 2 (*continued*)

The preprocessing phase is executed to generate shortest path trees from the origin node to all other nodes and from the destination node to all other nodes in the network for each objective (i.e., the two original objectives and the combined objective) for reducing the network size and accelerating the solution search progress. These added algorithmic components only polynomially increase the computational cost and memory requirement. It is well known that the computation of shortest path trees can be accomplished by the standard label-setting algorithm (e.g., Dijkstra's algorithm), which is of polynomial-time complexity. The labeling phase requires no more than an additional feasibility check at each node when it is

Step 3.2 **for** each $(i,j) \in E$ **do**

  if $c_l(i) + c_{l,ij} + c'_l(j) \geq \max\left(z_{1,l}^{(k,q)}, z_{2,l}^{(k,q)}\right)$, $\exists l = 1, 2$ **then**

   remove arc (i,j) from network G;

  otherwise

   if $d(i) + \sum_l w_l^p c_{l,ij} + d'(i) \geq U$ **then**

    remove arc (i,j) from network G;

   otherwise

    if $c_l(i,\Delta(i)) + c_{l,ij} + c'_l(j,\Delta'(j)) \leq \max\left(z_{1,l}^{(k,q)}, z_{2,l}^{(k,q)}\right)$, $\forall l = 1, 2$ and the solution corresponding to the combination of $\Delta(i)$, (i,j), and $\Delta'(j)$ is not identical to any one of the efficient paths corresponding to $\mathbf{z}_1^{(k,q)}$ and $\mathbf{z}_2^{(k,q)}$ **then**

     set $U = d(i) + \sum_l w_l^p c_{l,ij} + d'(i)$ and record the corresponding solution;

    end

   end

  end

 end

end

end

phase 2 labeling

begin

Step 0 initialize $S := \emptyset$, $\bar{S} := V$;

set $\bar{c}_l(i) := \infty$, $\forall i \in V$, $\forall l = 1, 2$, $\bar{d}(r) := \infty$;

set $\bar{c}_l(r) := 0$, $\forall l = 1, 2$, $\bar{d}(r) := 0$;

Step 1 **while** $s \notin \bar{S}$ or $\bar{c}_l(j) := \infty$, $\forall j \in \bar{S}$ **do**

 move node i from $\bar{S}$ to S for i satisfying $d(i) = \min_{j \in \bar{S}} d(j)$;

 for each $(i,j) \in A(i)$ **do**

  if $\bar{c}_l(i,\bar{\Delta}(i)) + c_{l,ij} + c'_l(j) < \max\left(z_{1,l}^{(k,q)}, z_{2,l}^{(k,q)}\right)$, $\forall l = 1, 2$ **then**

   if $\bar{d}(i) + \sum_l w_l^q c_{l,ij} + d'(j) < U$ **then**

    if $\bar{d}(i) + \sum_l w_l^q c_{l,ij} < \bar{d}(j)$ **then**

     set $\bar{c}_l(j,\bar{\Delta}(j)) := \bar{c}_l(i,\bar{\Delta}(i)) + c_{l,ij}$, $\forall l = 1, 2$;

     set $\bar{d}(j) := \bar{d}(i) + \sum_l w_l^q c_{l,ij}$;

     set $pred(j) := i$;

     if $\bar{c}_l(i,\bar{\Delta}(i)) + c_{l,ij} + c'_l(j,\Delta'(j)) < \max\left(z_{1,l}^{(k,q)}, z_{2,l}^{(k,q)}\right)$, $\forall l = 1, 2$ **then**

      $U = \bar{d}(i) + \sum_l w_l^q c_{l,ij} + d'(j)$;

     end

    end

   end

  end

 end

end

Step 2 retrieve optimal solution $\mathbf{x}$ and its attribute set $\mathbf{z}$;

if $s \in S$ **then**

 conclude $\mathbf{x}$ is an efficient solution and return it to the main routine as $\mathbf{x}^*$;

otherwise

 conclude no efficient solution is found and return to the main routine;

end

(b) Approximate doubly-constrained label-setting algorithm

Fig. 2 (*continued*)

scanned and updated, compared to the standard label-setting algorithm. Therefore, the computation time spent in this phase is also limited to the polynomial-time level.

It must be noted that though the two-phase label-setting algorithm ensures the feasibility of solutions, it does not always guarantee the solution optimality. Suboptimality might occur when the optimal path is dominated by an infeasible path (which has at least one of the single objectives violating the corresponding constraint) at some node j when the status of the unfeasible path reaching this node is feasible, i.e., $\bar{c}_l(i, \overline{\Delta}(i)) + c_{l,ij} + c\prime_l(j) < \max(z_{1,l}, z_{2,l})$. In other words, it is possible that an optimal path is dominated by an unfeasible path at an intermediate stage of the dynamic updating process of the algorithm. As we will see from the result of the numerical examples later on, however, this suboptimality occurrence may be limited to a small possibility when the above two-phase label-setting algorithm is used to tackle the parameterized, doubly-constrained shortest path problem.

There are a few reasons contributing to the reduction of the solution suboptimality of the biobjective shortest path problem. First, if the optimal solution to the parameterized, doubly-constrained problem is an extreme Pareto-optimal solution to the biobjective problem, the algorithm can find this optimal solution for sure; in fact, in this case, removal of the objective constraints does not affect the optimization result. This fact shows that the suboptimality condition never occurs with an extreme Pareto-optimal solution. Second, in many cases, the optimal solution, which is either an extreme or nonextreme Pareto-optimal solution, is identified by the preprocessing phase of the approximate label-setting algorithm, which does not introduce any solution suboptimality. Third, a suboptimal solution to the parameterized, doubly-constrained problem is possibly still a Pareto-optimal solution to the biobjective problem. In this case, the suboptimality does not actually affect the solution quality of the original biobjective problem. Fourth, even if a suboptimal solution is not a Pareto-optimal solution, it may be detected when the algorithm finds a new Pareto-optimal solution that dominates it. In this case, the suboptimal solution, or the dominated solution, can be discarded from the Pareto-optimal set. Fifth, as we will show later, in a multiobjective shortest path problem with three or more objectives, which can be decomposed to a number of biobjective problems, any suboptimal solution obtained by solving one of the biobjective problems may be detected by solving other biobjective problems, which further reduces the possibility of suboptimal solutions entering the Pareto-optimal solution set of the original multiobjective problem. The synthesis of these factors limits the possibility of solution suboptimality to a very low level, as will be shown in a performance comparison between the developed approximate algorithm and an exact algorithm in Section 5.

Following the discussions above, we highlight below a few algorithmic features of the proposed approximate constrained label-setting algorithm.

Property 1. *The approximate constrained label-setting algorithm is a polynomial-time algorithm, where its worst-case complexity is $O(|V|^2)$, given $|V|^2 > |E|$.*

Property 2. *The approximate constrained label-setting algorithm finds a nondominated solution in such a way: if the two parameter-generating Pareto-optimal solutions are extreme solutions, the generated Pareto-optimal solution, if exists, is extreme or nonextreme; if one of the parameter-generating Pareto-optimal solutions is a nonextreme solution, the generated Pareto-optimal solution, if exists, is nonextreme.*

Property 3. *The constrained label-setting algorithm guarantees finding all extreme Pareto-optimal solutions.*

According to Property 3, we know that the constrained parametric method is at least as good as the generic parametric method in terms of the Pareto-optimal solution completeness.

3.3. An illustrative example of biobjective shortest path problems

A biobjective shortest path example problem is illustrated here to demonstrate the effectiveness of the proposed parametric algorithm in searching for both extreme and nonextreme Pareto-optimal solutions. The network and its arc attributes of this example are shown in Fig. 3a, where the origin r and destination s are nodes 1 and 9, respectively. We intend to find all Pareto-optimal solutions from r and s.

At the initial stage, given the initial parameter sets, $\mathbf{w}_1 = (\varepsilon, 1-\varepsilon)$ and $\mathbf{w}_2 = (1-\varepsilon, \varepsilon)$, where we set $\varepsilon = 0.01$, the first two Pareto-optimal solutions are identified by simply applying a label-setting method (e.g., Dijkstra's algorithm); then, iteratively, the perpendicular method is used to generate new parameter sets and the constrained label-setting method to find new Pareto-optimal solutions. The problem is solved in 7 iterations, which find 5 Pareto-optimal solutions, including 4 extreme solutions and 1 nonextreme solution. The solution search process and result are recorded in Table 2. It can be simply proved by enumeration that this is a complete Pareto-optimal solution set to the given illustrative biobjective problem. All the solutions are plotted in the objective space in Fig. 3b, graphically showing their vector dominance relationship.

The effectiveness of the approximate constrained label-setting algorithm searching for both extreme and nonextreme Pareto-optimal solutions can be further justified by checking some example iterations. Here we choose iterations 1, 4 and 7, which represent the results of the algorithm finding an extreme solution, a nonextreme solution, and no solution, respectively.

Specifically, in iteration 1, given $z_1 = (z_{1,1}, z_{1,2}) = (33, 21)$ and $z_2 = (z_{2,1}, z_{2,2}) = (25, 29)$, and $c_1(s, \Delta(s)) = 28$ and $c_2(s, \Delta(s)) = 23$, it is clear that $c_l(s, \Delta(s)) < \max(z_{1,l}^{(k,q)}, z_{2,l}^{(k,q)}), \forall l = 1, 2$. So the algorithm concludes at step 1.2 of the preprocess-

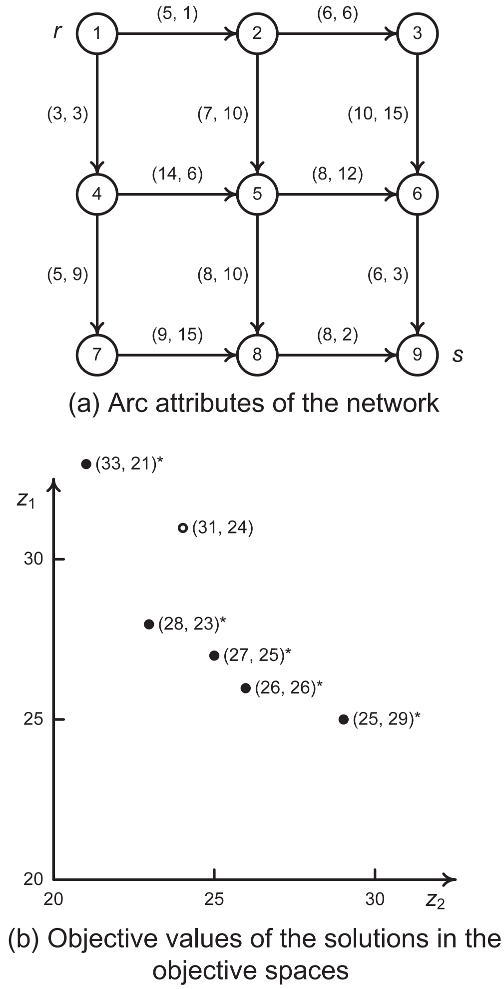


Fig. 3. An illustrative example of biobjective shortest path problems.

Table 2
Solution search process and results of the constrained parametric algorithm for the illustrative biobjective shortest path example.

Iteration	Parameter set	Objective upper bound	Pareto-optimal solution	Pareto-optimal objective vector
0	$\mathbf{w}_1 = (0.01, 0.99)$	N/A	$\mathbf{x}_1^* : 1-4-5-8-9$	$\mathbf{z}_1^* = (33, 21)$
	$\mathbf{w}_2 = (0.99, 0.01)$		$\mathbf{x}_2^* : 1-4-7-8-9$	$\mathbf{z}_2^* = (25, 29)$
1	$\mathbf{w}_3 = (0.5, 0.5)$	$z_{1,max}^{(3)} = 33,\quad z_{2,\max}^{(3)} = 29$	$\mathbf{x}_3^* : 1-2-5-8-9$	$\mathbf{z}_3^* = (28, 23)$
2	$\mathbf{w}_4 = (0.29, 0.71)$	$z_{1,\max}^{(4)} = 33,\quad z_{2,\max}^{(4)} = 23$	$\mathbf{x}_4^*$: N/A	$\mathbf{z}_4^*$ N/A
3	$\mathbf{w}_5 = (0.67, 0.33)$	$z_{1,\max}^{(5)} = 28,\quad z_{2,\max}^{(5)} = 29$	$\mathbf{x}_5^* : 1-2-5-6-9$	$\mathbf{z}_5^* = (26, 26)$
4	$\mathbf{w}_6 = (0.6, 0.4)$	$z_{1,\max}^{(6)} = 28,\quad z_{2,\max}^{(6)} = 26$	$\mathbf{x}_6^* : 1-2-3-6-9$	$\mathbf{z}_6^* = (27, 25)$
5	$\mathbf{w}_7 = (0.75, 0.25)$	$z_{1,\max}^{(7)} = 26,\quad z_{2,\max}^{(7)} = 29$	$\mathbf{x}_7^*$: N/A	$\mathbf{z}_7^*$ N/A
6	$\mathbf{w}_8 = (0.67, 0.33)$	$z_{1,\max}^{(8)} = 28,\quad z_{2,\max}^{(8)} = 25$	$\mathbf{x}_7^*$: N/A	$\mathbf{z}_7^*$ N/A
7	$\mathbf{w}_9 = (0.5, 0.5)$	$z_{1,\max}^{(9)} = 27,\quad z_{2,\max}^{(9)} = 26$	$\mathbf{x}_7^*$: N/A	$\mathbf{z}_7^*$ N/A

ing phase that the solution $\mathbf{x}(\Delta(s))$ is the optimal solution of the current constrained shortest-path problem and is a Pareto-optimal solution to the original biobjective problem. In iteration 4, the algorithm in turn removes arcs 1–4, 4–5, 4–7 and 7–8 at step 3.1 and arc 5–8 at step 3.2 of the preprocessing phase and obtain the upper bound of the objective function $U = 26.2$; the algorithm then repeats executing the preprocessing phase in the reduced network, which results in a further removal of arcs 2–5, 5–6, and 8–9 at step 3.1 and a removal of arcs 2–3 and 3–6 at step 3.2; finally, the optimal solution is identified at step 1.3 of the preprocessing phase by retrieving the solution corresponding to the upper bound of the objective function. As for iteration 7, the algorithm first removes arcs 2–3, 3–6, 4–7, 7–8, 1–4 and 4–5 at step 1.3 of the preprocessing phase and

then removes arcs 5–8 at step 3.2; after such a network reduction, the network G contains only a single path from the origin node and destination node. Next, the second time of executing the preprocessing phase found at step 2.2 that this constrained shortest path problem has no feasible solution. From these example iterations, we can see the effectiveness of the preprocessing phase of the constrained label-setting algorithm in finding optimal solutions and excluding the existence of feasible solutions. A thorough check into the solution process reveals that in all its seven iterations of performing the constrained label-setting algorithm for this trivial example problem, the second phase—the labeling phase—is not invoked at all, which means that all Pareto-optimal solutions identified by the approximate algorithm are exact solutions. In our extensive test, as we will discuss later on, it is found that in the parametric solution framework the preprocessing phase can optimally solve the parameterized, constrained shortest path problem without resorting to the labeling phase in many problem cases.

4. Decomposition scheme for multiobjective problems

While it is analytically and geometrically proved that the constrained parametric method described above solves the biobjective shortest path problem (in an optimal or suboptimal manner) optimally or near optimally, it seems difficult to apply a similar algorithmic procedure to an n-objective problem when $n \geqslant 3$. The following insights provide a simple strategy to decompose a multiobjective problem to a number of biobjective problems. The complete Pareto-optimal solution set of the original multiobjective problem is a union set of the Pareto-optimal sets of all these biobjective problems.

4.1. Problem decomposition and solution combination

Theorem 2. *A solution to an n-objective minimization problem* $\mathbf{x}$ *with its objective set* $\mathbf{z}(\mathbf{x}) = (z_1(\mathbf{x}), \ldots, z_l(\mathbf{x}), \ldots, z_n(\mathbf{x}))$ *is Pareto-optimal (i.e., n-nondominated) if and only if there exists at least such an m-dimensional partial objective set* $\mathbf{z}'(\mathbf{x}) = (\ldots, z_l(\mathbf{x}), \ldots)$, *where* $2 \leqslant m \leqslant n$ *and* $z_l(\mathbf{x}) \in \mathbf{z}(\mathbf{x})$, $\forall l$, *that the solution is Pareto-optimal in terms of its m objectives (i.e., m-nondominated), unless there is another solution that has the exactly same objective values in terms of the m objectives.*

Proof. Suppose that a solution $\mathbf{x}$ is Pareto-optimal in terms of the m-dimensional partial objective set $\mathbf{z}'(\mathbf{x})$ and no other solution has the same objective values in terms of the m objectives. According to Definition 1, its Pareto-optimality to the optimization problem is not changed when we consider it in n objectives instead of m objectives, since $\mathbf{z}'(\mathbf{x}) \subseteq \mathbf{z}(\mathbf{x})$. The sufficiency is proved.

Suppose that a solution $\mathbf{x}$ is Pareto-optimal in terms of the n-dimensional full objective set $\mathbf{z}(\mathbf{x})$. In other words, there does not exist any other feasible solution that dominates $\mathbf{x}$, or for any other solution $\mathbf{y}$, there exists at least one objective l so that $z_l(\mathbf{y}) > z_l(\mathbf{x})$. Thus, we can construct an m-dimensional partial objective set $\mathbf{z}'(\mathbf{x}) = (\ldots, z_l(\mathbf{x}), \ldots)$, where m is an integer number between 2 and n and $z_l(\mathbf{x})$ is an element of $\mathbf{z}'(\mathbf{x})$. This condition ensures that solution $\mathbf{x}$ is Pareto-optimal in terms of these m objectives. The necessity is proved. □

In case a Pareto-optimal solution in terms of an m-dimensional partial objective set has the same objective values as another Pareto-optimal solution in terms of the same partial objective set, however, it is not necessary for this or the other solution to be Pareto-optimal in terms of the complete n-dimensional objective set. For example, given two solutions $(5, 10, 15)$ and $(5, 10, 10)$ of a three-objective minimization problem, the two solutions have the same objective values $(5, 10)$ in terms of the first two objectives. Let us assume that these two solutions are Pareto-optimal in terms of the first two objectives if no other solution dominates them. Given such a partial Pareto-optimal setting, however, we can still find that the first solution is not Pareto-optimal in terms of the complete three-objective set because it is dominated by the second solution if we consider the third objective. From this example, it is readily known that if two or more solutions have the same objective values in terms of a partial objective set, their Pareto-optimality may not be fully determined until we compare them in terms of a higher-dimensional objective set; if their objective values are still the same in terms of this higher-dimensional objective set, we then need to add more objectives into the objective set for comparison, and in turn repeat this process, if needed, until reaching the full objective set.

Following the theorem presented above, we extend our conclusion into the following corollary without proof, which provides a useful algorithmic device to reduce the dimensionality of any multiobjective optimization problem and the computational complexity of a high-dimensional multiobjective optimization problem.

Corollary 1. *The Pareto-optimal solution set of a multiobjective optimization problem in terms of its n-dimensional full objective set is a union set of the Pareto-optimal solutions of the same problem in terms of each of its m-dimensional partial objective set, excluding those solutions that have the same objective values as other solutions in terms of an m-dimensional objective set but are dominated by these other solutions in terms of a higher-dimensional objective set, where* $2 \leqslant m \leqslant n$.

Based on the conclusions above, a problem decomposition and solution combination method can be constructed as follows to tackle a higher-dimensional multiobjective optimization problem by solving a number of lower-dimensional problems. All solution points of each lower-dimensional problem in its objective space are a projection of those of the

```
                algorithm multiobjective problem decomposition and solution combination method
                begin
Step 1              for i_1 := 1 to n − m + 1 do
                        create a one-dimensional objective set by adding the i_1th objective into a blank set;
                        for i_2 := i_1 + 1 to n − m + 2 do
                            create a two-dimensional objective set by adding the i_2th objective into the i_1th one-
                            dimensional objective set containing i_1;
                            …;
                            for i_m := i_{m−1} + 1 to n do
                                create an m-dimensional objective set by adding the i_mth objective into the
                                i_{m−1}th (m − 1)-dimensional objective set containing i_1, …, i_{m−1};
                                solve the m-objective optimization problem;
                                eliminate m-nondominated solutions that are not n-nondominated;
                            end
                            …;
                        end
                    end
Step 2              find the union set of m-nondominated solutions of all m-objective optimization problems and
                    eliminate any dominated solutions in term of the n-dimensional objective set;
                end
```

Fig. 4. Problem decomposition and solution combination method for multiobjective optimization problems.

higher-dimensional problem in the higher-dimensional objective space. Under this projection scheme, we know that if we decompose an n-objective optimization problem into a set of m-objective problems, where $2 \leqslant m \leqslant n$, the complete set contains $n!/m!(n-m)!m$-objective problems. In this case, since we have developed an efficient (approximate) algorithm for biobjective shortest path problems, any multiobjective shortest path problem can be indirectly treated by approximating a number of easier biobjective problems. For instance, we can decompose a triobjective problem into three biobjective problems, and a four-objective problem into six biobjective problems.

The major algorithmic steps of this problem decomposition and solution combination method are depicted in Fig. 4. For illustration, we also show the process of applying this method for solving an example triobjective shortest path problem below. Of course, the resulting biobjective problems by decomposition are all tackled by the constrained parametric method developed in the above text. Just because of the possible solution suboptimality introduced by the constrained parametric method for biobjective shortest path problems, a dominance check is needed when the Pareto-optimal solutions of these biobjective problems are combined into the union set.

4.2. An illustrative example of triobjective shortest path problems

A triobjective shortest path problem is formed by adding a third arc attribute to the biobjective problem described above, as shown in Fig. 5. We decompose this triobjective problem into three biobjective problems, in terms of z_1 and z_2, z_3 and z_2, and z_1 and z_3, respectively, and apply the constrained parametric method to solve them. As shown in Table 3, solving the problem with z_1 and z_2 results in 4 partial Pareto-optimal solutions; solving the problem with z_3 and z_2 results in 2 Pareto-optimal solutions; and solving the problem with z_1 and z_3 results in 3 Pareto-optimal solutions. A union set of these partial Pareto-optimal solutions results in 6 Pareto-optimal solutions, which constitute the full Pareto-optimal solution set of this illustrative triobjective shortest path problem. Fig. 5 graphically exhibits the 3-dimensional vector dominance relationship projected into three 2-dimensional objective spaces. By enumeration, we know that the union set does not contain any non-Pareto-optimal solution and these solutions exhaust all Pareto-optimal solutions of the example triobjective problem.

5. Computational results

The illustrative examples presented above demonstrate the basic algorithmic ideas and operations of the approximate constrained parametric algorithm and the decomposition scheme, but have not provided any insights about the algorithm's computational performance in large-scale applications. To get some sense on the solution performance of the developed parametric search and problem decomposition procedures, we set up the following numerical experiments. The purpose of this set of experiments is twofold: (1) examine the solution generation behavior implied by the parametric algorithm and (2) evaluate the solution completeness and efficiency of the parametric algorithm in a range of networks with different problem dimensions, sizes, and attribute sets. For a comparative assessment, we simultaneously implemented a multiobjec-

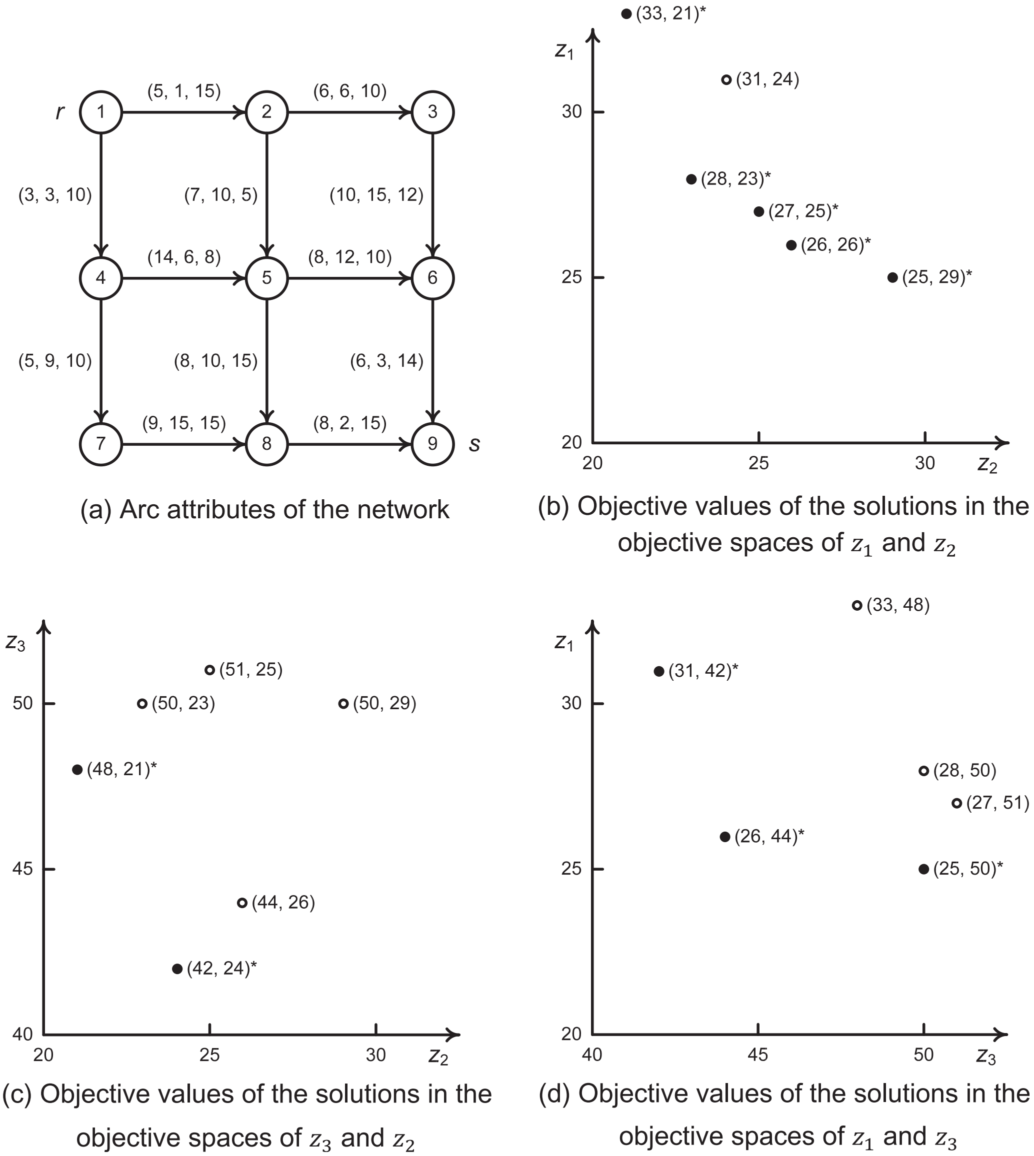


Fig. 5. An illustrative example of triobjective shortest path problems.

tive labeling algorithm as a benchmark in all the problems instances. It should be emphasized, however, that this comparison is rather illustrative than comprehensive, since a number of other existing exact and approximate multiobjective shortest path algorithms in the literature are not implemented here. This remains a future task to the authors.

5.1. Biobjective problems

A set of 10 random grid networks are generated with their size ranging from 78 to 3,920 nodes and from 240 to 12,480 arcs (see Table 4). Use of grid networks is to mimic the typical topology of urban transportation networks. For illustration, an example of the random grid topology used in this study is presented in Fig. 6. The ratio of number of arcs to number of nodes in these grid networks is around 3. For each of these grid networks, we generate 10 random arc attribute sets; each of these attribute sets includes a two-attribute vector for each arc in the network. Without loss of generality, we generate each single arc attribute independently and identically by a uniform distribution bounded by 0 and 100. Following such a random network and attribute generation mechanism, we obtain totally a set of 100 network-attribute scenarios for the biobjective problem set. In each scenario, we arbitrarily select the node at the top left corner as the origin and the node at the bottom right corner as the destination (see Fig. 6).

Table 3
Pareto-optimal solutions of the illustrative triobjective shortest path example.

Solution	Pareto-optimal objective vector w.r.t. (z_1, z_2)	Pareto-optimal objective vector w.r.t. (z_3, z_2)	Pareto-optimal objective vector w.r.t. (z_1, z_3)	Pareto-optimal objective vector w.r.t. (z_1, z_2, z_3)
1–2–3–6–9	(27, 25)	–	–	(27, 25, 50)
1–2–5–6–9	(26, 26)	–	(26, 44)	(26, 26, 44)
1–2–5–8–9	(28, 23)	–	–	(28, 23, 50)
1–4–5–6–9	–	(42, 24)	(31, 42)	(31, 24, 42)
1–4–5–8–9	(33, 21)	(48, 21)	–	(33, 21, 48)
1–4–7–8–9	(25, 29)	-	(25, 50)	(25, 29, 50)

Table 4
List of biobjective shortest problem instances.

Network	Number of nodes	Number of arcs	Label-correcting method		Constrained parametric method	
			Average number of nondominated paths	Average computation time (s)	Average number of nondominated paths	Average computation time (s)
1	78	240	9.4	0.009	9.2	0.053
2	280	880	23.1	0.117	22.7	0.297
3	462	1456	29.3	0.247	27.9	0.561
4	688	2716	40.7	0.707	40.5	1.031
5	1061	3360	49.5	3.119	47.8	1.991
6	1512	4800	80.9	7.973	76.8	4.770
7	2044	6496	115.0	14.382	110.1	8.340
8	2822	8976	142.5	42.364	136.1	15.660
9	3348	10,656	156.0	70.160	146.3	22.899
10	3920	12,480	211.5	163.628	198.7	36.571

The benchmark algorithm we selected for the comparative evaluation is Brumbaugh-Smith and Shier's (1989) biobjective label-correcting method. This is an exact solution algorithm that can find the full set of Pareto-optimal paths. To the authors' best knowledge, this algorithm (and its extensions) represents the most efficient class among all available biobjective labeling methods reported in the literature. We coded both the constrained parametric algorithm and the label-correcting algorithm in C++ and compiled it using Microsoft C/C++ Optimizing Compiler. All the computational runs presented below were conducted on a Windows-based computer with a Core 2 Duo 2.40 GHz CPU and 2 GB RAM.

The comparative evaluation is conducted in terms of two performance measures: (1) solution quality (or completeness), i.e., the number of generated true Pareto-optimal solutions; (2) solution efficiency, i.e., the computation time used to

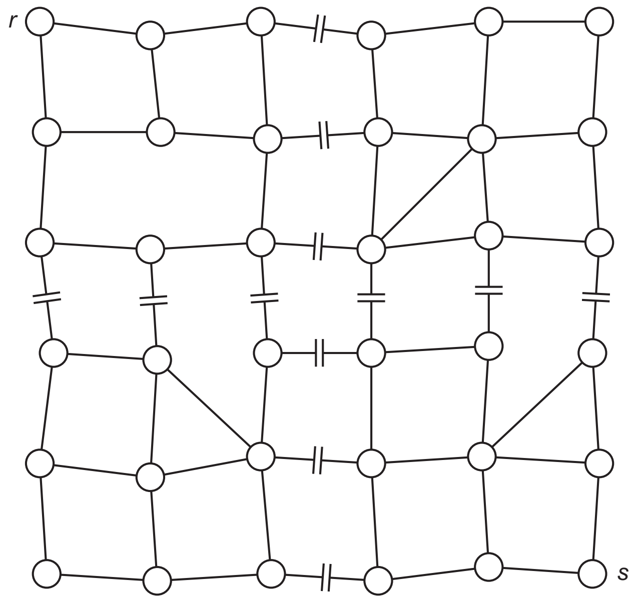
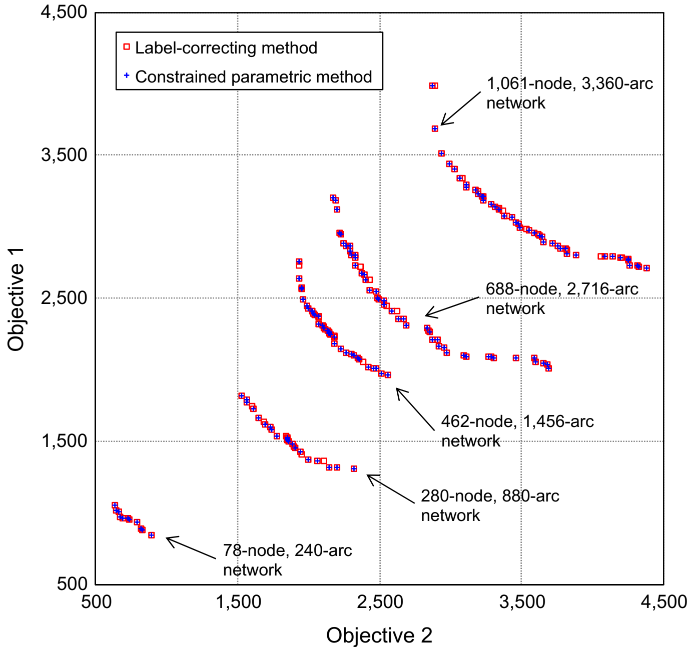
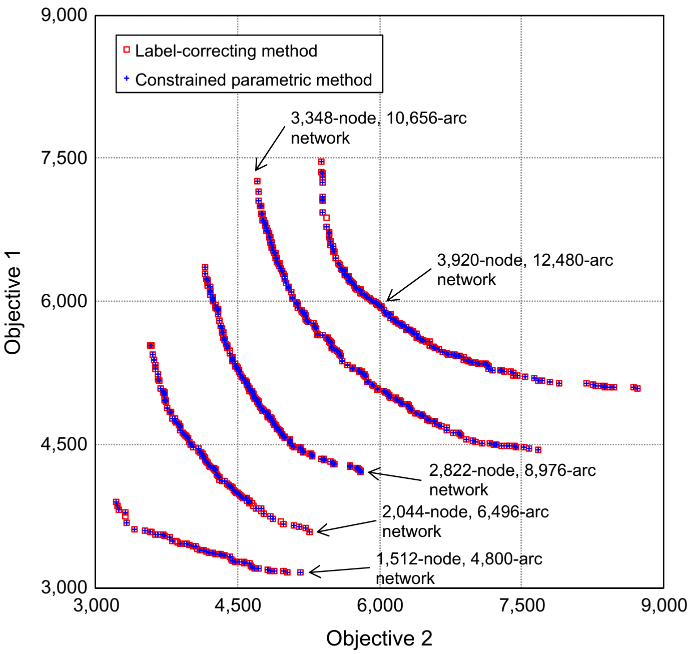


Fig. 6. Topology of the random grid networks.



(a) Example networks with 78-1,061 nodes and 240-3,360 arcs



(b) Example networks with 1,512-3,920 nodes and 4,800-12,480 arcs

Fig. 7. Pareto-optimal solution profiles of the example biobjective shortest path problem instances.

generate the Pareto-optimal solution set. Due to the requirement of evaluating the solution quality, we selected an exact rather than any approximate algorithm as the benchmark.

In Fig. 7, we present and compare in the objective space the Pareto-optimal solution distributions of a set of example problems solved by the constrained parametric algorithm and the label-correcting algorithm. The example problems are

ten instances selected from the generated network-attribute scenarios, each of which is from one grid network. From Fig. 7, we can see that the Pareto-optimal solution profiles generated by the constrained parametric algorithm exhibit a very good match to the corresponding ones by the label-correcting algorithm, where a large number of nonextreme Pareto-optimal solutions are included in these solution profiles.

It should be noted that the order of generated Pareto-optimal solutions is determined by the parameter generation process, which in general proceeds in a distributed rather than concentrated manner over the whole parameter range. Because of this reason, we can use a subset of earliest generated Pareto-optimal solutions by the parametric algorithm to represent or approximate the solution diversity before the whole Pareto-optimal solution set is obtained. In contrast, the labeling algorithm cannot determine any Pareto-optimal path until its label-updating process is done for the whole network. In fact, it generates all the Pareto-optimal solutions at the end of the search process. Due to this difference in solution generation behavior, it is expected that the parametric algorithm may be preferred to the labeling algorithm if the complete Pareto-optimal solution set is not required, since the former can generate a set of "representative" Pareto-optimal solutions in a relatively faster manner, especially for networks of large size.

A more comprehensive comparison is made for the parametric algorithm and the labeling algorithm by synthetically reviewing all the computational results in the biobjective case. In Fig. 8, we plot the average number of true Pareto-optimal paths generated by both the algorithms over all 10 attribute sets of each network. Since the labeling algorithm guarantees the generation of the full set of Pareto-optimal solutions, which sets a solution quality standard, the relative solution quality of the parametric algorithm can be indicated by its *solution identification rate*, which is here defined as the ratio of the number of generated Pareto-optimal solutions to the number of Pareto-optimal solutions in the full set. From Fig. 8, it is seen that while the number of Pareto-optimal solutions in the full set increases over the network size, the solution identification rate of the parametric algorithm approximately ranges from 85% to 100% over all network sizes.

While the parametric algorithm may not exhaust all Pareto-optimal paths, the computational advantage of this method is favored by its appealing polynomial-time complexity. In fact, the parametric algorithm can perform more efficiently than the labeling algorithm at the cost of missing a small part of Pareto-optimal solutions. To see this, we depict the average computation time over the network size for both the parametric algorithm and the labeling algorithm in Fig. 9. As a detailed comparison, we present these computation time results on two scales. Specifically, a small scale is used to cover network instances with the number of arcs less than 3500, while a large scale is used for presenting the full range of network sizes. It is found that when the network size is relatively small (e.g., grid networks with approximately less than 2500 arcs), the parametric algorithm runs slower than the labeling algorithm (see Fig. 9a); when the network size is larger than this threshold, the parametric algorithm outperforms the labeling algorithm and its computation time saving becomes more significant as the network size increases (see Fig. 9b). In the problem instances of the largest network size, for example, the average computation time of the parametric algorithm is about 26.6 s, which is merely about 30% of that of the labeling algorithm, 90.6 s.

5.2. Triobjective problems

In the triobjective case, we generate an alternative set of 5 random grid networks with their node numbers ranging from 78 to 1061 and arc numbers ranging from 240 to 3360 arcs (see Table 5). Again, all of these grid networks are constructed with a similar topology to the sample network in Fig. 6. Following the same attribute generation rule as the biobjective case, we generate 10 random arc attribute sets for each of the triobjective networks here, where each arc is assigned with three uniformly generated attributes ranging from 0 to 100. Thus, we get 50 network-attribute scenarios in total for the triobjective case.

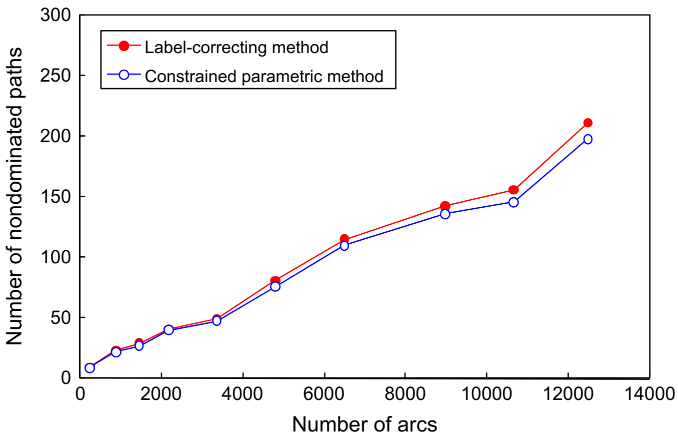


Fig. 8. Biobjective shortest path problems: number of Pareto-optimal paths over network size.

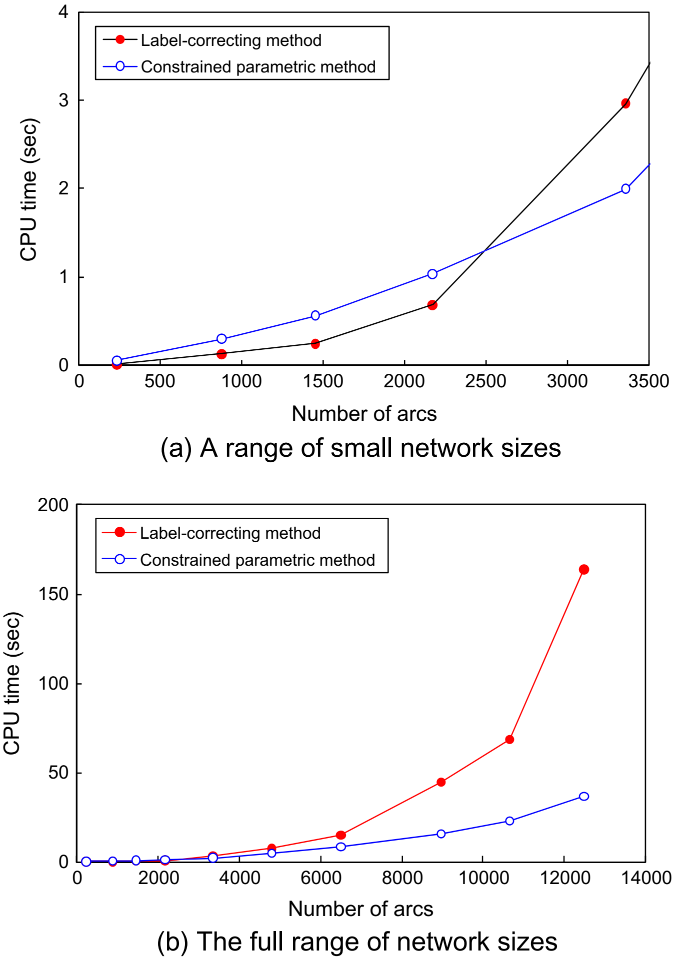


Fig. 9. Biobjective shortest path problems: computation time over network size.

Table 5
List of triobjective shortest path problem instances.

Network	Number of nodes	Number of arcs	Label-correcting method		Constrained parametric method	
			Average number of nondominated paths	Average computation time (s)	Average number of nondominated paths	Average computation time (s)
1	78	240	22.6	0.012	22.3	0.214
2	280	880	38.8	0.923	38.7	1.128
3	462	1456	61.5	6.528	57.2	2.243
4	688	2716	85.5	24.803	81.9	7.526
5	1061	3360	133.7	183.230	125.9	27.965

A set of comparative analyses similar to those applied to the biobjective case are also used here to examine the results of the triobjective problems. Different from the biobjective parametric algorithm, solving a triobjective shortest path problem under the parametric solution framework requires the solutions of three biobjective problems (obtained through the decomposition scheme). The biobjective label-correcting algorithm, in contrast, can be directly applied to solve triobjective problems with only slight modifications in its dominance check procedure. Fig. 10 depicts the number of triobjective Pareto-optimal solutions generated by the parametric algorithm and the labeling algorithm over different network sizes. It shows that the parametric algorithm consistently identifies approximately 87–100% of the Pareto-optimal solutions over different problem instances. Its average identification rate is slightly higher than that of the parametric algorithm in the biobjective case.

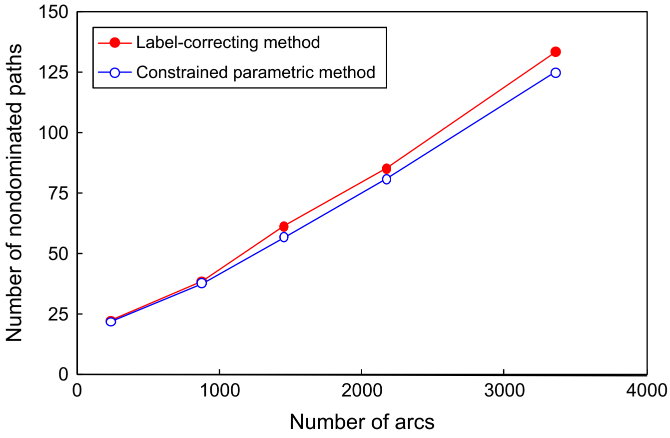


Fig. 10. Triobjective shortest path problems: number of Pareto-optimal paths over network size.

Fig. 11 compares the computation costs consumed by the parametric algorithm and the labeling algorithm in the triobjective case. Similarly, it shows that, in terms of computation time, the parametric algorithm underperforms the labeling algorithm for network instances of relatively small size (i.e., networks with less than 800–900 arcs), but surpasses the

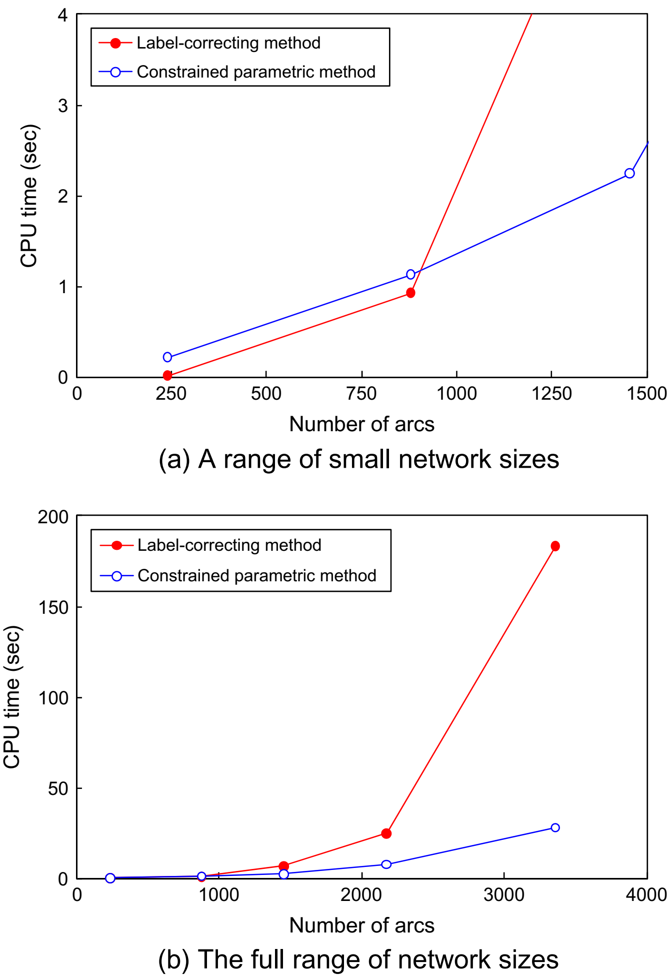


Fig. 11. Triobjective shortest path problems: computation time over network size.

labeling algorithm when the network size arises beyond this threshold. It is apparent that when the network size increases, the computation time required by the labeling algorithm increases drastically, while the parametric algorithm increases its computation time at a quite moderate rate.

The different increasing patterns of computation costs of the two algorithms are rooted from their different algorithmic designs. It is well known that the computation cost of the labeling algorithm is primarily determined by the number of dominance checks during the label-updating process. The computation cost of the parametric algorithm is a function of the number of Pareto-optimal paths as well as the search process of additional origin-based and destination-based shortest paths in its preprocessing phase.

When the network size is small, the number of potential Pareto-optimal paths arriving at an immediate node and the number of dominance checks are accordingly small. In this case, the dominance check process of a multiobjective labeling algorithm is not significantly more costly than its single-objective counterpart. On the other hand, the parametric method generates Pareto-optimal paths on an individual basis, each of which is a set of single-objective shortest path problems. When the network size is relatively large, the number of potential dominance checks at intermediate nodes exponentially increases in its worst case, hence the resulting computation cost of the labeling algorithm. The computation cost of the parametric method, on the other hand, increases with the number of actual Pareto-optimal paths and the cost of solving single-objective path problems. In our case, the former increases moderately with the increase in network size, while the latter is clearly a polynomial-time process. Evidently, these different computation cost results reflect the polynomial-time and exponential-time complexities of the two algorithms.

6. Concluding remarks and future directions

Optimal routing problems with respect to multiple objectives pose important theoretical and practical implications. They not only arise as stand-alone applications in their own right but also often serve as subroutines in more complex multiobjective network flow problems. This paper focuses on the development of an approximate constrained parametric algorithm that is capable of potentially identifying extreme and nonextreme biobjective Pareto-optimal paths in a polynomial-time manner. A decomposition method is also supplemented as an algorithmic tool to convert a multiobjective shortest path problem into a set of biobjective problems, so that the biobjective parametric algorithm can be indirectly used for solutions of multiobjective problems.

The generation of the entire Pareto-optimal solution set is not practical or necessary in most of real-world multiobjective applications (Current et al., 1990; Rennen et al., 2011). Multiobjective optimal routing problems are not an exclusive case. In many cases, we only need to find a small set of Pareto-optimal routes that can cover representative tradeoffs between conflicting objectives. Quickly identifying a representative Pareto-optimal set is especially important for real-time, en-route decision making. The parametric algorithm is ideally used in time-constrained environments, in that its search process can be terminated earlier as needed, as long as the number or density of solution points over the Pareto-optimal profile satisfies the prespecified requirement. Due to its individual path generation mechanism, the parametric algorithm has much less memory requirement than labeling algorithms and other types of multiobjective shortest path methods. This feature also allows the parametric algorithm to be implemented via parallel computing techniques, which can significantly reduce the computation time, especially for networks of large size.

This paper computationally compares the solution completeness and efficiency of the approximate constrained parametric algorithm and a well known benchmark algorithm—the label-correcting algorithm—in both biobjective and triobjective cases. While the comparison result is encouraging and promising, this comparison is rather illustrative than comprehensive and is based on rather synthetic than realistic problems. It is more interesting to know how the constrained parametric algorithm performs when it competes with other approximate algorithms, in terms of solution completeness and efficiency, as well as their solution-generating behaviors, using an extensive set of real-world problems. This computational study remains as a future task. On the other hand, it should be realized that the developed biobjective parametric algorithm and multiobjective decomposition scheme are only applicable to Pareto-optimal routing problems of the *min-sum* type. How to modify it to accommodate other types of Pareto-optimal routing problems, such as *min–max*, *max–min*, or *max–sum* problems, is an important task to address in the future. In addition, use of the parametric search idea for multiobjective optimal routing problems in dynamic and stochastic networks also poses some impending research questions.

Acknowledgements

The authors like to express their appreciation to an anonymous reviewer for his or her constructive suggestions for improving the paper. The research is partially supported by TxDOT and CAMPO through the Network Modeling Center at the University of Texas at Austin. However, the authors claim the full responsibility for the facts and the accuracy of the data presented in the paper.

References

Aneja, Y.P., Nair, K.P.K., 1979. Bicriterion transportation problem. Management Science 25 (1), 73–78.
Aneja, Y.P., Aggarwal, V., Nair, K.P.K., 1983. Shortest chain subject to side constraints. Networks 13 (2), 295–302.

Beasley, J.E., Christofides, N., 1989. An algorithm for the resource constrained shortest path problem. Networks 19 (4), 379–394.
Bellman, R., 1957. Dynamic Programming. Princeton University Press, Princeton, NJ.
Boffey, B., 1995. Multiobjective routing problems. Top 3 (2), 167–220.
Bonzon, P., 1970. Necessary and sufficient conditions for dynamic programming of combinatorial type. Journal of Association for Computing Machinery 17 (4), 675–682.
Brown, T.A., Strauch, R.E., 1965. Dynamic programming in multiplicative lattices. Journal of Mathematics and Applications 12 (2), 364–370.
Brumbaugh-Smith, J., Shier, D., 1989. An empirical investigation of some bicriterion shortest path algorithms. European Journal of Operational Research 43 (2), 216–224.
Carlyle, W.M., Royset, J.O., Wood, R.K., 2008. Lagrangian relaxation and enumeration for solving constrained shortest-path problems. Networks 52 (4), 256–270.
Chang, T.S., Nozick, L.K., Turnquist, M.A., 2005. Multiobjective path finding in stochastic dynamic networks with application to routing hazardous materials shipments. Transportation Science 39 (3), 383–399.
Clímaco, J.C.N., Martins, E.Q.V., 1982. A bicriterion shortest path algorithm. European Journal of Operational Research 11 (4), 399–404.
Cohon, J.L., 1978. Multiobjective Programming and Planning. Academic Press, New York, NY.
Corley, H.W., Moon, I.D., 1985. Shortest path in networks with vector weights. Journal of Optimization Theory and Applications 46 (1), 79–86.
Coutinho-Rodrigues, J.M., Clímaco, J.C.N., Current, J.R., 1999. An interactive biobjective shortest path approach: searching for unsupported nondominated solutions. Computers and Operations Research 26 (6), 789–798.
Cox, R.G., 1984. Routing and Scheduling of Hazardous Materials Shipments: Algorithmic Approaches to Managing Spent Nuclear Fuel Transport. PhD thesis. School of Civil and Environmental Engineering, Cornell University, Ithaca, NY.
Current, J.R., Min, H., 1986. Multiobjective design of transportation networks: Taxonomy and annotations. European Journal of Operational Research 26 (1), 187–201.
Current, J.R., Marsh, M., 1993. Multiobjective transportation network design and routing problems: Taxonomy and annotation. European Journal of Operational Research 65 (1), 4–19.
Current, J.R., ReVelle, C.S., Cohon, J.L., 1990. An interactive approach to identify the best compromise solution for two objective shortest path problems. Computers and Operations Research 17 (2), 187–198.
Daellenbach, H.G., De Kluyver, C.A., 1980. Note on multiple objective dynamic programming. Journal of the Operational Research Society 31 (7), 591–594.
Desrochers, M., Soumis, F., 1988. A generalized permanent labeling algorithm for the shortest path problem with time windows. Information Systems and Operational Research 26 (3), 193–214.
Desrosiers, J., Dumas, Y., Solomon, M., Soumis, F., 1995. Time constrained routing and scheduling. In: Ball, M.O., Magnanti, T.L., Monma, C.L., Nemhauser, G.L. (Eds.), Handbook in Operations Research and Management Science, Network Routing, vol. 8. North-Holland, Amsterdam, pp. 35–139.
Dumitrescu, I., Boland, N., 2001. Algorithms for the weight constrained shortest path problems. International Transaction in Operations Research 8 (1), 15–29.
Dumitrescu, I., Boland, N., 2003. Improved preprocessing labeling and scaling algorithms for the weight-constrained shortest path problem. Networks 42 (3), 135–153.
Ehrgott, M., Gandibleux, X., 2000. A survey and annotated bibliography of multiobjective combinatorial optimization. OR Spektrum 22 (4), 425–460.
Ekrut, E., Tjandra, S.A., Verter, V., 2007. Hazardous materials transportation. In: Barnhart, C., Laporte, G., (Eds.), Transportation Handbooks in Operations Research and Management Science, vol. 14, Elsevier, Amsterdam, The Netherlands, pp. 539–621.
Fruhwirth, B., Burkard, R.E., Rote, G., 1989. Approximation of convex curves with application to the bicriterial minimum cost flow problem. European Journal of Operational Research 42 (3), 326–338.
Garey, M., Johnson, D., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco, CA.
Guerriero, F., Musmanno, R., 2001. Label correcting methods to solve multicriteria shortest path problems. Journal of Optimization Theory and Applications 111 (3), 589–613.
Handler, G.Y., Zang, I., 1980. A dual algorithm for the constrained shortest path problem. Networks 10 (4), 293–309.
Hansen, P., 1979. Bicriterion path problems. In: Fandel, G., Gal, T. (Eds.), Multiple Criteria Decision Making: Theory and Application, Lecture Notes in Economics and Mathematics Systems, vol. 17. Springer-Verlag, Heidelberg, Germany, pp. 109–127.
Hassin, R., 1992. Approximation schemes for the restricted shortest path problem. Mathematics of Operations Research 17 (1), 36–42.
Henig, M.I., 1985. The shortest path problem with two objective functions. European Journal of Operational Research 25 (2), 281–291.
Ignizio, J.P., 1976. Goal Programming and Extensions. Lexington Books, Lanham, MD.
Jaffe, J.M., 1984. Algorithms for finding paths with multiple constraints. Networks 14 (1), 95–116.
Jaumard, B., Semet, F., Vovor, T., 1996. A Two-Phase Resource Constrained Shortest Path Algorithm for Acyclic Graphs. Les Cahier du GERAD, G-96-48, Université de Montréal, Montréal, QC.
Lawler, E.L., 1976. Combinatorial Optimization: Networks and Matroids. Holt Rinehart and Winston, New York, NY.
Lorenz, D.H., Raz, D., 2001. A simple efficient approximation scheme for the restricted shortest path problem. Operations Research Letters 28 (5), 213–219.
Loui, R., 1983. Optimal paths in graphs with stochastic or multidimensional weights. Communications of the Association for Computing Machinery 26 (9), 670–676.
Martins, E.Q.V., 1984. On a multicriteria shortest path problem. European Journal of Operational Research 16 (2), 236–245.
Meng, Q., Lee, D.H., Cheu, R.L., 2009. Multiobjective vehicle routing and scheduling problem with time window constraints in hazardous material transportation. Journal of Transportation Engineering 131 (9), 699–707.
Mote, J., Murthy, I., Olson, D.L., 1991. A parametric approach to solving bicriterion shortest path problems. European Journal of Operational Research 53 (1), 81–92.
Nielsen, L.R., Andersen, K.A., Pretolani, D., 2003. Bicriterion shortest hyperpaths in random time-dependent networks. Journal of Management Mathematics 14 (3), 271–303.
Nozick, L.K., List, G.F., Turnquist, M.A., 1997. Integrated routing and scheduling in hazardous materials transportation. Transportation Science 31 (3), 200–215.
Opasanon, S., Miller-Hooks, E., 2006. Multicriteria adaptive paths in stochastic time-varying networks. European Journal of Operational Research 173 (1), 72–91.
Özpeynirci, Ö., Köksalan, M., 2010. An exact algorithm for finding extreme supported nondominated points of multiobjective mixed integer programs. Management Science 56 (12), 2302–2315.
Przybylski, A., Gandibleux, X., Ehrgott, M., 2010a. A recursive algorithm for finding all nondominated extreme points in the outcome set of a multiobjective integer programme. INFORMS Journal on Computing 22 (3), 371–386.
Przybylski, A., Gandibleux, X., Ehrgott, M., 2010b. A two phase method for multi-objective integer programming and its application to the assignment problem with three objectives. Discrete Optimization 7 (3), 149–165.
Raith, A., Ehrgott, M., 2009. A comparison of solution strategies for biobjective shortest path problems. Computers and Operations Research 36 (4), 1299–1331.
Reeves, D.S., Salama, H.F., 2000. A distributed algorithm for delay-constrained unicast routing. IEEE/ACM Transactions on Networking 8 (2), 239–250.
Rennen, G., van Dam, E.R., den Hertog, D., 2011. Enhancement of sandwich algorithms for approximating higher-dimensional convex Pareto sets. INFORMS Journal on Computing 23 (4), 493–517.
Ribeiro, C., Minoux, M., 1985. A heuristic approach to hard constrained shortest path problems. Discrete Applied Mathematics 10 (2), 125–137.
Robbins, J.C., 1983. Routing Hazardous Material Shipments. PhD thesis. Department of Geography, Indiana University, Bloomington, IN.

Sastry, V.N., Janakiraman, T.N., Mohideen, S.I., 2003. New algorithms for multi-objective shortest path problem. Opsearch 40 (4), 278–298.
Serafini, P., 1986. Some considerations about computational complexity for multi-objective combinatorial problems. In: Jahn, J., Krabs, W. (Eds.), Recent Advances and Historical Development of Vector Optimization, Lecture Notes in Economics and Mathematical Systems, vol. 294. Springer-Verlag, Berlin, pp. 222–232.
Skriver, A.J.V., Andersen, K.A., 2000. A label correcting approach for solving bicriterion shortest-path problems. Computers and Operations Research 27 (6), 507–524.
Sun, Q., Langendörfer, H., 1998. A new distributed routing algorithm for supporting delay-sensitive applications. Computer Communications 21 (6), 572–578.
Tung, T.C., Chew, K.L., 1988. A bicriterion Pareto-optimal path algorithm. Asia–Pacific Journal of Operational Research 5 (2), 166–172.
Tung, T.C., Chew, K.L., 1992. A multicriteria Pareto-optimal path algorithm. European Journal of Operational Research 6 (2), 203–209.
Turnquist, M.A., 1987. Routes schedules and risks in transporting hazardous materials. In: Lev, B. et al. (Eds.), Strategic Planning in Energy and Natural Resources. Elsevier Science, Amsterdam, The Netherlands, pp. 289–302.
Ulungu, E.L., Teghem, J., 1994. Multiobjective combinatorial optimization problems: a survey. Journal of Multi-Criteria Decision Analysis 3 (2), 83–104.
Vincke, P., 1974. Problèmes multicritères. Cahiers du Centre D'Etudes de Recherche Opérationnelle 16, 425–439.
Warburton, A., 1987. Approximation of Pareto optima in multiple-objective shortest-path problems. Operations Research 35 (1), 70–79.
White, D.J., 1982. The set of efficient solutions for multiple objective shortest path problems. Computers and Operations Research 9 (2), 101–107.
White, D.J., 1990. A bibliography on the applications of mathematical programming multiple-objective methods. The Journal of the Operational Research Society 41 (8), 669–691.
Xiao, Y., Thulasiraman, K., Xue, G., Jüttnet, A., 2005. The constrained shortest path problem: algorithmic approaches and an algebraic study with generalization. International Journal of Graphs and Combinatorics 2 (2), 63–86.