Journal of Global Optimization **15:** 405–417, 1999.

Polynomial algorithms for a class of minimum rank-two cost path problems

TAKAHITO KUNO
Institute of Information Sciences and Electronics, University of Tsukuba, Japan

(Accepted in revised form 28 July 1999)

Abstract. In this paper, we develop two algorithms for finding a directed path of minimum rank-two monotonic cost between two specified nodes in a network with n nodes and m arcs. Under the condition that one of the vectors characterizing the cost function f is binary, one yields an optimal solution in $O(n^3)$ or $O(nm \log n)$ time if f is quasiconcave; the other solves any problem in $O(nm + n^2 \log n)$ time.

Key words: Global optimizationl, low rank monotonicity, polynomial algorithm, shortest path, Dijkstra's algorithm.

1. Introduction

A number of global optimization problems encountered in real-world applications have some special structures which enable us to design efficient algorithms [7]. One of the most favorable structures is the *low rank monotonicity* studied by Tuy et al. [11,13,14]. The nonlinearity of any rank k monotonic function f is located in a subspace of dimension k even if f is defined on a subset of much higher dimensional space than k. Functions of this kind appear in multiplicative programming [10,18], facility location [15], multilevel programming [17] and certain variants of minimum concave-cost network flows, for which even polynomial algorithms have been developed [6,9,16]. Especially in multiple objective decision making, they play an important role [3,5]. In fact, when a decision maker has k linear objectives $c^1x, \dots, c^kx$ without a common scale, a handy approach to optimizing them simultaneously is to optimize a rank k monotonic function such as $f(x) = \prod_{i=1}^{k} (c^ix + \alpha_i)$ or $f(x) = \max\{\alpha_i c^i x \mid i = 1, \dots, k\}$ for some constants α_i's.

In this paper, we consider a minimum rank-two cost path problem, i.e., a problem of finding a directed path which minimizes a rank-two monotonic cost function f between two specified nodes in a given network with n nodes and m arcs. Recently, in-car navigation systems using artificial satellites have made it possible to find a way to a destination without road maps. The present systems, however, only provide several candidate routes from which a driver must make a selection while driving. Therefore, efficient algorithms for minimizing rank-two monotonic functions will be helpful in reducing the driver's burden.

The organization of the paper is as follows. In Section 2, we will describe the problem formally and show that it is a $\mathcal{NP}$-hard problem. In Sections 3 and 4, we will concentrate on a class that one of the vectors characterizing the rank-two monotonic cost function f is binary. We will develop two algorithms for solving the class: one yields an optimal solution in $O(n^3)$ or $O(nm \log n)$ arithmetic operations and $O(n^2)$ evaluations of f if the cost function f is quasiconcave; the other solves any problem in this class in $O(nm + n^2 \log n)$ arithmetic operations and $O(n)$ evaluations of f. In Section 5, we will briefly discuss an application of these algorithms to the general class of problems.

2. Minimum rank-two cost path problem

Let $G = (N, A)$ be a graph consisting of a set N of n nodes and a set A of m directed arcs. Our purpose is to determine a directed path of minimum cost from a specified node s to another specified node t in G. When the number of times the path traverses each arc $(i, j) \in A$ is x_{ij}, it costs $f(x)$, where $x \in Z^m$ is the vector of x_{ij}'s. We assume that the cost function $f : R^m \to R$ is continuous on some open convex set D, which includes the set X of all $x \geqslant 0$ satisfying

$$\sum_{\{j|(i,j)\in A\}} x_{ij} - \sum_{\{j|(j,i)\in A\}} x_{ji} = \begin{cases} 1 & \text{for } i = s \\ -1 & \text{for } i = t \\ 0 & \text{for each } i \in N\backslash\{s, t\}. \end{cases} \tag{2.1}$$

We further assume that f is *rank-two monotonic* on D with respect to two nonnegative vectors c^1 and $c^2 \in Z^m$ [12,18]. Namely,

(i) the vectors c^1 and c^2 are linearly independent;
(ii) if $x, y \in D$ and $c^k(x - y) \geqslant 0$ for $k = 1, 2$, then $f(x) \geqslant f(y)$.

As will be seen later, f can be a convex function; but the class also involves nonconvex functions such as multiplicative functions $f_1(x) = (c^1x + \alpha_1)(c^2x + \alpha_2)$ on $D_1 = \{x \in R^m | c^kx + \alpha_k > 0,\ k = 1, 2\}$ and fractional functions $f_2(x) = c^1x/(\alpha_3 - c^2x)$ on $D_2 = \{x \in R^m | c^1x > 0,\ \alpha_3 - c^3x > 0\}$. For other examples of nonconvex f, see a recent textbook of structured nonconvex optimization by Konno, Thach and Tuy [11].

We call the problem described above a *minimum rank-two cost path problem*, which can be formulated as follows:

(MR2P) minimize $\{f(x) | x \in X \cap Z^m\}$.

Under conditions (i) and (ii), an optimal solution x^* to (MR2P) is given by an elementary path P if nodes s and t are connected. For suppose that P contains a directed cycle C. Let

$$y_{ij} = \begin{cases} x^*_{ij} - 1 & \text{if } (i, j) \in C \\ x^*_{ij} & \text{otherwise.} \end{cases}$$

Then we have $c^k(x^* - y) = \sum_{(i,j)\in C} c^k_{ij} \geqslant 0$ for $k = 1, 2$, which implies that $f(x^*) \geqslant f(y)$. The cost does not rise even if C is discarded from P.

Problem (MR2P), though simple looking, is intractable from the viewpoint of worst-case complexity; and in fact it belongs to the class $\mathcal{NP}$-hard. To see this, let us consider the following recognition problem:

SHORTEST WEIGHT-CONSTRAINT PATH (SWCP) [4]
INSTANCE: Graph $G = (N, A)$, positive length $l_{ij} \in Z$, positive weight $w_{ij} \in Z$ for each $(i, j) \in A$, specified nodes $s, t \in N$, positive integers K, W.
QUESTION: Is there a path in G from s to t with total weight W or less and total length K or less?

The recognition version of the 0-1 knapsack problem, well known to be $\mathcal{NP}$-complete, can reduce in polynomial time to this problem (see, e.g., Ahuja et al. [1]); and hence (SWCP) is an $\mathcal{NP}$-complete problem.

Choosing any instance of (SWCP), let us define a convex function:

$$f_3(x) = \max\{lx - K, wx - W\},$$

where l and w are the vectors of l_{ij}'s and w_{ij}'s, respectively. If l and w are linearly dependent, the instance is equivalent to an ordinary shortest path problem and can be solved in polynomial time; therefore, we can assume condition (i) for l and w without loss of generality. Moreover, we can see that f_3 satisfies condition (ii) on R^m with respect to l and w. In other words, f_3 is a rank-two monotonic function. The instance has the 'yes' solution if and only if G contains an s-t path with nonpositive $f_3(x)$, which can be verified by solving (MR2P) with $f = f_3$. Consequently, we have

PROPOSITION 2.1. *Problem (MR2P) is $\mathcal{NP}$-hard.*

In the rest of this paper, we concentrate on a class of (MR2P) where all the nonzero components of c^1 or c^2 are the same value. Since f is rank-two monotonic with respect to $\alpha_1 c^1$ and $\alpha_2 c^2$ for any positive α_k's, we can assume either of the vectors to be binary. We then show that this class can be solved in polynomial time. Certainly, it covers only part of (MR2P), but is substantial in practical applications. For example, in navigation systems, we may wish to find a route that is short in length and simultaneously has few intersections to a destination. We will have a reasonable route by minimizing a rank-two monotonic function, say $(dx + \alpha_1)(ex + \alpha_2)$ or $\max\{dx, \alpha_3 ex\}$, where α_k's are appropriate constants, e is the vector of ones, and each component of d represents the distance between two adjoining intersections.

Let $\{A_0, A_+\}$ be a partition of the arc set A, i.e., $A_0 \cap A_+ = \emptyset$ and $A_0 \cup A_+ = A$. In the sequel, we assume that

$$c_{ij}^2 = \begin{cases} 0 & \text{for each } (i, j) \in A_0 \\ 1 & \text{for each } (i, j) \in A_+ \end{cases} \tag{2.2}$$

Note that $A_+ \neq \emptyset$; otherwise, condition (i) is not satisfied. Also, we assume for simplicity that network G contains a directed path from node s to node t. We can easily check it by solving an ordinary shortest path problem. Under these conditions, we will discuss the following two cases:

Case 1. f is a rank-two monotonic and continuous quasiconcave function on D;

Case 2. f is a rank-two monotonic and general continuous function on D.

3. Parametric cost algorithm for Case 1

We first show that (MR2P) satisfying condition (2.2) can be solved in polynomial time if the cost fucntion f is quasiconcave on D, i.e., for any $x, y \in D$, we have

$$f[(1-\lambda)x + \lambda y] \geqslant \min\{f(x), f(y)\} \text{ for any } \lambda \in [0, 1]. \tag{3.1}$$

The functions f_1 and f_2 given in Section 2 satisfy this condition on D_1 and D_2, respectively [2].

Whenever f satisfies (3.1), we can omit the integrality constraint $x \in Z^m$ and write the problem simply as follows:

$$\text{minimize}\{f(x) | x \in X\}. \tag{3.2}$$

The minimum of f is achieved at some vertex x^* of the polyhedron X. The total unimodularity of the incidence matrix of G guarantees that x^* is an integral vector and provides an optimal s-t path [1]. We also have the following regardless of condition (2.2):

THEOREM 3.1. *If f is quasiconcave on D, there is some constant $\lambda \geqslant 0$ such that any optimal solution to a problem*

$$[PC(\lambda)] \textit{ minimize}\{c^1x + \lambda c^2 x | x \in X\}$$

is an optimal solution to (3.2).

Proof. See Theorems 9.1 and 9.2 in Konno et al. [11]. □

This theorem holds true even for the problem without network structures so long as f is rank-two monotonic with respect to c^1, c^2 and bounded from below on X. Tuy and Tam [18] have used it and proposed a parametric simplex algorithm for minimizing a rank-two monotonic quasiconcave function over a general polytope. Since (3.2) is a special case of their problem, we can solve it in the same way as in Tuy and Tam [18].

Note that, in our case, [PC(λ)] is a shortest path problem with nonnegative arc length $c^1+\lambda c^2$ for any $\lambda \geqslant 0$. We can compute an optimal solution $x(\lambda)$ to [PC(λ)] in $O(m+n \log n)$ time, using Dijkstra's algorithm [1]. Let

$$X^* = \{x \in R^m | x = x(\lambda),\ \lambda \geqslant 0\}.$$

We see from Theorem 3.2 that an optimal solution to (3.2) is given by

$$x^* \in \arg\min\{f(x) | x \in X^*\}.$$

It will be time-consuming to obtain the whole of X^* if we use Dijkstra's algorithm to compute $x(\lambda)$ for each $\lambda \geqslant 0$. We can, however, accomplish it in polynomial time using algorithms by Karp and Orlin [8]. To solve parametric shortest path problems just like [PC(λ)], they have developed two algorithms: an $O(n^3)$ algorithm based upon dynamic programming and an $O(nm \log n)$ network simplex algorithm. Both generate a partition $\{I^1, \ldots, I^r\}$ of the interval $[0, \infty)$ and a set $\{P^1, \ldots, P^r\}$ of s-t paths in G such that P^k is a shortest path, with respect to the arc length $c^1 + \lambda c^2$, for all $\lambda \in I^k$. Using either of them as a subroutine, we can tailor the algorithm by Tuy and Tam for problem (3.2).

algorithm PARAMETRIC_COST
begin
  using one of the algorithms in [8], compute a partition $\{I^1, \ldots, I^r\}$ of the interval $[0, \infty)$ and a set $\{P^1, \ldots, P^r\}$ of paths in G such that P_k is a shortest path from node s to node t for all $\lambda \in I^k$;
  $v := +\infty$;
  for $k = 1, \ldots, r$ **do begin**
    let x^k denote the vector corresponding to P^k;
    if $f(x^k) < v$ **then** $v := f(x^k)$ and $x^* := x^k$
  end
end;

For each $k = 1, \ldots, r$, the vector x^k is an optimal solution to [PC(λ)] for all $\lambda \in I^k$. Since $\cup_{k=1}^r I^k = [0, +\infty)$, the set $\{x^1, \ldots, x^r\}$ can be thought of as X^*; and x^* refers to an optimal solution to (3.2) at the end of the algorithm. The number r of paths $P^1, \ldots, P^r$ is known to be at most n^2 [8]. Therefore, PARAMETRIC_COST requires $O(n^2)$ evaluations of f in addition to $O(n^3)$ or $O(nm \log n)$ arithmetic operations. This result is rather satisfactory compared with those for ordinary shortest path problems. Since network simplex algorithms are very efficient in practice, PARAMETRIC_COST would be as well when it uses the $O(nm \log n)$ subroutine. Unfortunately, however, the algorithm PARAMETRIC_COST only works on problems with quasiconcave cost functions. Unless the cost function is quasiconcave, the algorithm can fail even in a toy problem.

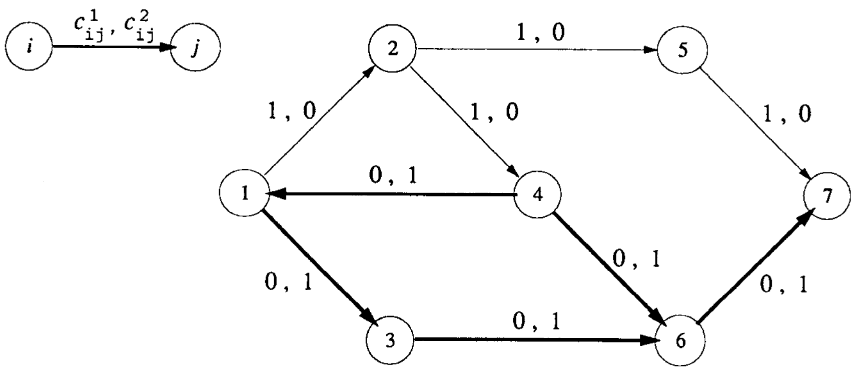

Figure 1. The network G of problem (3.3).

EXAMPLE 3.1. Consider the following problem with a rank-two convex cost function:

$$\left|\begin{array}{lll} \text{minimize} & f(x) = \max\{c^1x, c^2x\} & \\ \text{subject to} & x_{12} + x_{13} - x_{41} = 1, & x_{24} + x_{25} - x_{12} = 0 \\ & x_{36} - x_{13} = 0, & x_{41} + x_{46} - x_{24} = 0 \\ & x_{57} - x_{25} = 0, & x_{67} - x_{36} - x_{46} = 0 \\ & -x_{57} - x_{67} = -1 & \\ & x_{ij} : \text{ nonnegative integer for each } (i, j), & \end{array}\right. \tag{3.3}$$

where

$$\begin{array}{l} c^1_{12} = c^1_{24} = c^1_{25} = c^1_{57} = 1, \;\; c^1_{13} = c^1_{36} = c^1_{41} = c^1_{46} = c^1_{67} = 0 \\ c^2_{12} = c^2_{24} = c^2_{25} = c^2_{57} = 0, \;\; c^2_{13} = c^2_{36} = c^2_{41} = c^2_{46} = c^2_{67} = 1. \end{array}$$

Figure 1 shows the network G associated with this problem, where the fine and bold lines represent the arcs in A_0 and A_+, respectively.

It is easy to see from Figure 1 that an optimal path is $P^* = (1, 2, 4, 6, 7)$ of cost $\max\{c^1_{12} + c^1_{24}, c^2_{46} + c^2_{67}\} = 2$; but the algorithm PARAMETRIC_COST only generates

$$\begin{array}{lll} I^1 = [0, 1], & P^1 = (1, 3, 6, 7), & \max\{0, c^2_{13} + c^2_{36} + c^2_{67}\} = 3 \\ I^2 = [1, +\infty), & P^2 = (1, 2, 5, 7), & \max\{c^2_{12} + c^2_{25} + c^2_{57}, 0\} = 3 \end{array}$$

and misses P^*.

This example suggests that we have to device another algotithm to solve more general class of (MR2P).

4. Parametric right-hand-side algorithm for Case 2

Without assuming f to be quasiconcave, let us consider the class of (MR2P)

$$\text{minimize}\{f(x)|x \in X \cap Z^m\} \tag{4.1}$$

which satisfies condition (2.2) and contains an s-t path in the underlying network G.

To solve (4.1), we again introduce a parameter $\lambda \geqslant 0$ but in a way different from [PC(λ)]:

$$\text{minimize}\{f(x)|x \in X \cap Z^m,\ c^2x = \lambda,\ \lambda \geqslant 0\}. \tag{4.2}$$

Since the value of c^2x is always nonnegative on X, this problem is equivalent to (4.1). We also see from condition (ii) of the rank-two monotonicity that once the value of λ is fixed, (4.2) reduces to a shortest path problem with a side constraint

$$[\text{PR}(\lambda)]\ \text{minimize}\{c^1x|x \in X \cap Z^m,\ c^2x = \lambda\}.$$

Let $x(\lambda)$ be an optimal solution to [PR(λ)] if it exists, and let $g(\lambda) = f[x(\lambda)]$, where $f[x(\lambda)]$ is understood to be $+\infty$ if [PR(λ)] has no optimal solutions. Then (4.2) amounts to a minimization of the univariate function:

$$\text{minimize}\{g(\lambda)|\lambda \geqslant 0\}.$$

While an optimal path of (MR2P) is elementary and contains at most $n-1$ arcs, the path corresponding to $x(\lambda)$ contains at least λ arcs. Therefore, to locate a minimum point λ^* of g, we need only to solve [PR(λ)] for each $\lambda \in [0, n-1]$. An optimal solution $x(\lambda^*)$ to [PR(λ^*)] solves the target problem (4.1).

Whether f is quasiconcave or not, this approach never misses an optimal solution to (4.1); but it seems no good from the computational viewpoint – for shortest path problems with a side constraint are in general intractable, as is (SWCP). Under condition (2.2), however, we can show that the total computational time needed in this approach is polynomial in n and m.

4.1. AUXILIARY NETWORK

Let $d^\lambda(i, j)$ denote the distance, with respect to the arc length c^1, from node i to node j along a shortest path that contains exactly λ arcs in A_+. If such a path does not exist, then $d^\lambda(i, j) = +\infty$. Naturally, $d^\lambda(s, t)$ is equal to the optimal value $g(\lambda)$ of [PR(λ)].

Now, suppose that a path $P = (s = i_0, i_1, \dots, i_{r+1}, i_r = j)$ provides $d^\lambda(i, j) < +\infty$ for $\lambda \geqslant 1$. Let (i_q, i_{q+1}) be the last A_+ arc that we pass when going along the path P from node s. In other words, a subpath $(i_{q+1}, \dots, i_r)$ of P consists of only arcs in A_0. We then see that paths $(i_0, i_1, \dots, i_q)$ and $(i_{q+1}, \dots, i_r)$ provide

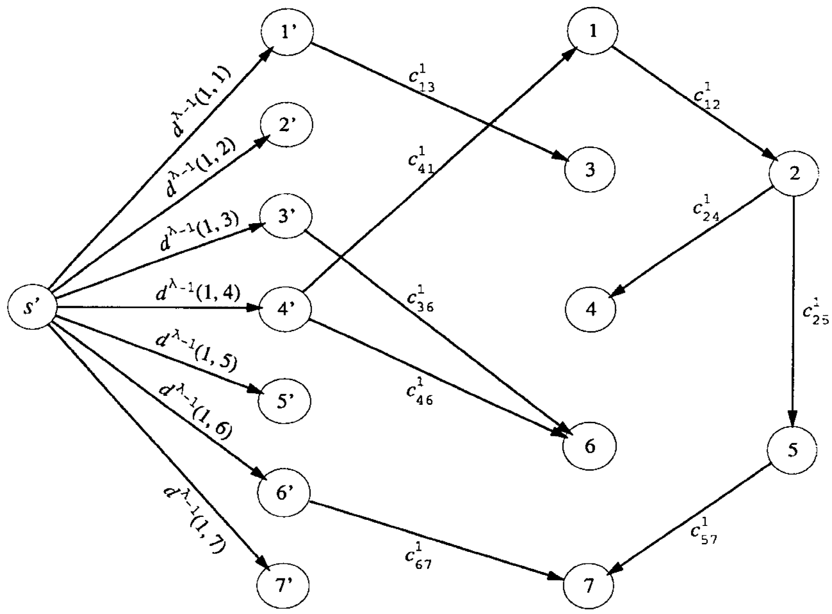

Figure 2. The auxiliary network $G(\lambda)$ associated with problem (3.3).

$d^{\lambda-1}(s, i_q)$ and $d^0(i_{q+1}, j)$, respectively. Otherwise, P cannot be a shortest path containing exactly λ arcs in A_+. The following is an immediate consequence:

LEMMA 4.1. *For each* $\lambda \geqslant 1$ *we have*

$$d^\lambda(s, j) = \min\{d^{\lambda-1}(s, k) + c^1_{kl} + d^0(l, j) | (k, l) \in A_+\} \text{ for every } j \in N. \tag{4.3}$$

Using this relationship, we can successively generate $d^0(s, t)$, $d^1(s, t), \ldots, d^{n-1}(s, t)$, among which is the minimum value of g, i.e., the length of an s-t path optimal for [PR(λ^*)] and hence for (4.1). To carry out this in a systematic and efficient way, we introduce an auxiliary network $G(\lambda)$.

Given $d^{\lambda-1}(s, j)$ with $\lambda \geqslant 1$, we construct $G(\lambda)$ from $G = (N, A_0 \cup A_+)$ as follows. We make a copy N' of the original node set N, and replace each arc $(i, j) \in A_+$ by an arc (i', j) of length c^1_{ij} from node $i' \in N'$ to node $j \in N$. We further introduce an artificial node s' and connect it and each node $i' \in N'$ with an artificial arc (s', i') of length $d^{\lambda-1}(s, i)$. Figure 2 illustrates the resulting network when we apply this transformation to the network in Figure 1.

It follows from the above construction of $G(\lambda)$ that any directed path from node s' to node $j \in N$ consists of three parts: the first part is an artificial arc (s', k') of length $d^{\lambda-1}(s, k)$; the second is an arc (k', l) of length c^1_{kl} substituting for an arc $(k, l) \in A_+$; and the third is a directed path from node k to node j in a subgraph (N, A_0) of G. Hence, from Lemma 4.1, the value of $d^\lambda(s, j)$ is given by the shortest path distance from node s' to node $j \in N$ in $G(\lambda)$. Since the length of each arc in $G(\lambda)$ is nonnegative, we can apply Dijkstra's algorithm to $G(\lambda)$ in order to compute $d^\lambda(s, j)$ for every $j \in N$.

4.2. DESCRIPTION OF THE ALGORITHM

We are ready to present the algorithm for solving problem (4.1).

algorithm PARAMETRIC_RHS
begin
  deterine the shortest path distance $d^0(s, j)$ from node s to each node $j \in N$ in the network (N, A_0) with arc length c^1;
  if $d^0(s, j) < +\infty$ **then** let P_j^0 denote the path of length $d^0(s, j)$;
  if $d^0(s, t) < +\infty$ **then**
    let x^* denote the vector corresponding to P_t^0 and $v := f(x^*)$;
  else $v := +\infty$
  for $\lambda = 1, \dots, n-1$ **do begin**
    construct the auxiliary network $G(\lambda)$;
    determine the shortest path distance $d^\lambda(s, j)$ from node s' to each node $j \in N$ in $G(\lambda)$;
    if $d^\lambda(s, j) < +\infty$ **then begin**
      replace the first two arcs (s', k') and (k', l') in the path length $d^\lambda(s, j)$ by $P_k^{\lambda-1}$ and $(k, l) \in A_+$;
      let P_j^λ denote the resulting path in G
    end;
    if $d^\lambda(s, t) < +\infty$ **then begin**
      let x^k denote the vector corresponding to P_t^λ;
      if $f(x^k) < v$ **then** $x^* := x^k$ and $v := f(x^*)$
    end
  end
end;

THEOREM 4.2. *The algorithm PARAMETRIC_RHS yields an optimal solution x^* to (4.1) in $O(nm + n^2 \log n)$ arithmetic operations and $O(n)$ evaluations of function f.*

Proof. The algorithm generates n distances $d^0(s, t)$, $d^1(s, t), \dots, d^{n-1}(s, t)$ between nodes s and t. Some of them might be $+\infty$; but at least one, as we have seen already, is provided by an optimal s-t path of (4.1). At the end of the algorithm, x^* refers to the s-t path.

Let us turn to the computational complexity. For every node $j \in N$, the distance $d^0(s, j)$ can be computed in $O(m + n \log n)$ arithmetic operations if Dijkstra's algorithm is applied to the network (N, A_0). In the λth iteration ($\lambda \geqslant 1$), there are two major tasks: to construct the auxiliary network $G(\lambda)$ and to compute $d^\lambda(s, j)$ for every node $j \in N$. The former requires $O(n + m)$ arithmetic operations in the first iteration; but afterwards only the lengths of at most m arcs need updating for each iteration. The latter can be done in $O(m + n \log n)$ arithmetic operations, using Dijkstra's algorithm, since $G(\lambda)$ contains $2n + 1$ nodes and $n + m$ arcs. Therefore, the total number of arithmetic operations is $O(nm + n^2 \log n)$. In addition to this, the algorithm evaluates f at most once for each iteration. Hence, the total number of evaluations of f is bounded by $O(n)$. □

EXAMPLE 4.2. Let us try to solve problem (3.3) in Example 3.1, using the algorithm PARAMETRIC_RHS.

To begin with, we determine the shortest path distance $d^0(1, j)$ from node $s = 1$ to each node $j = 1, \dots, 7$ in the subnetwork (N, A_0) (see Figure 1):

j	1	2	3	4	5	6	7
$d^0(1, j)$	0	1	$+\infty$	2	2	$+\infty$	3

Since a path $P_7^0 = (1.2, 5, 7)$ provides $d^0(1, 7) = 3 < +\infty$, we initialize the incumbent:

$$x_{ij}^* := \begin{cases} 1 & \text{if } (i, j) \in \{(1, 2),\ (2, 5),\ (5, 7)\} \\ 0 & \text{otherwise} \end{cases}$$

$$v := f(x^*) = \max\{c_{12}^1 + c_{25}^1 + c_{57}^1, 0\} = 3.$$

Then we proceed to the iteration process.

Iteration 1: We construct the auxiliary network $G(1)$, as shown in Figure 2, and determine the shortest path distance in it from node s' to node $j = 1, \dots, 7$:

j	1	2	3	4	5	6	7
$d^1(1, j)$	2	3	0	4	4	2	5

We obtain $P_7^1 = (1, 2, 4, 1, 2, 5, 7)$ from a path $(s', 4', 1, 2, 5, 7)$ of length $d^1(1, 7) = 5$ in $G(1)$; but it costs

$$f(x^1) = \max\{2c_{12}^1 + c_{24}^1 + c_{25}^1 + c_{57}^1,\ c_{41}^2\} = 5 > v.$$

Iteration 2: We update the auxiliary network $G(2)$ and determine the shortest path distance in it from node s' to each node $j = 1, \dots, 7$:

j	1	2	3	4	5	6	7
$d^2(1, j)$	4	5	2	6	7	0	2

We obtain $P_7^2 = (1, 2, 4, 6, 7)$ from a path $(s', 6', 7)$ of length $d^2(1, 7) = 2$ in $G(2)$; and it costs

$$f(x^2) = \max\{c_{12}^1 + c_{24}^1,\ c_{46}^2 + c_{67}^2\} = 2 < v.$$

Therefore, we update the incumbent:

$$x^*_{ij} := x^2_{ij} = \begin{cases} 1 & \text{if } (i, j) \in \{(1, 2),\ (2, 4),\ (4, 6),\ (6, 7)\} \\ 0 & \text{otherwise} \end{cases} \qquad (4.4)$$
$$v := f(x^2) = 2.$$

Iterations 3, ..., 6*:* In the same way as above, for each $\lambda = 3, \ldots, 6$, we compute $d^\lambda(1, j)$ for $j = 1, \ldots, 7$ and $f(x^k)$:

j	1	2	3	4	5	6	7	
$d^3(1, j)$	6	7	4	8	8	2	0	$f(x^3) = 3$
$d^4(1, j)$	8	9	6	10	10	4	2	$f(x^4) = 4$
$d^5(1, j)$	10	11	8	12	12	6	4	$f(x^5) = 5$
$d^6(1, j)$	12	13	10	14	14	8	6	$f(x^6) = 6$

Since $f(x^\lambda) > v$ for each $\lambda = 3, \ldots, 6$, the path P^2_7 is optimal; and (4.4) is an optimal solution to (3.3).

Thus, we succeeded in solving problem (3.3). Even if f is nonconvex in (3.3), the algorithm PARAMETRIC_RHS will generate the same sequence $d^\lambda(1, 7)$, $\lambda = 0, 1, \ldots, 6$; but possibly the output will be different from (4.4).

5. Concluding remark

In the previous sections, we have developed two parametric algorithms to solve a class of minimum rank-two cost path problem (MR2P) in which c^2, one of the vectors characterizing the rank-two monotonic cost function f, is a binary vector. The algorithm PARAMETRIC_COST yields an optimal solution in $O(n^3)$ or $O(nm \log n)$ arithmetic operations and $O(n^2)$ evaluations of f if the function f is quasiconcave. The algorithm PARAMETRIC_COST solves any problems in this class in $O(nm + n^2 \log n)$ arithmetic operations and $O(n)$ evaluations of f. Using these algorithms, we can solve the general class of (MR2P) in pseudopolynomial time.

For a given problem in (MR2P), we first transform the underlying network $G = (N, A)$ as follows (see also Figure 3). For each $(i, j) \in A$ with $c^2_{ij} > 1$, we install $c^2_{ij} - 1$ nodes on arc (i, j) and divide it into c^2_{ij} directed arcs. Let A_{ij} denote the set of arcs generated on arc $(i, j) \in A$. We then associate with each arc $(p, q) \in A_{ij}$ two numbers

$$\tilde{c}^1_{pq} = \begin{cases} c^1_{ij} & \text{if } p = 1 \\ 0 & \text{otherwise} \end{cases}, \ \tilde{c}^2_{pq} = 1. \qquad (5.1)$$

The resulting network, denoted by $\tilde{G}$, contains at most $n + C$ nodes and $m + C$ arcs, where $C = \sum_{(i,j)\in A} c^2_{ij}$. The original network G contains a path P from node

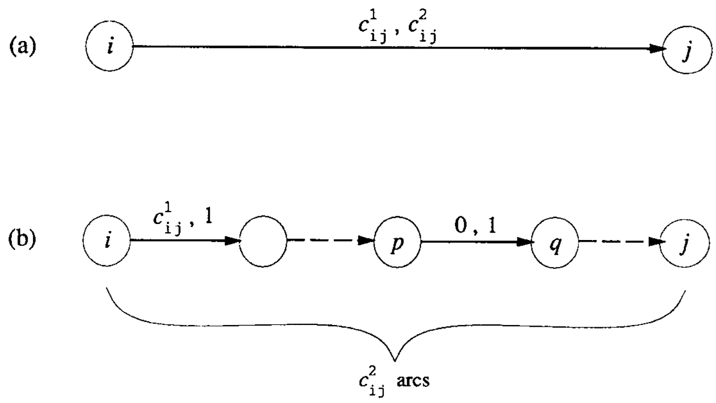

Figure 3. (a) arc (i, j) in the original network G; (b) arc set A_{ij} in the resulting network $\tilde{G}$.

s to node t if and only if $\tilde{G}$ contains a path $\tilde{P}$ between the same nodes. For each arc $(i, j) \in P$, the path $\tilde{P}$ contains all the arcs in A_{ij}. Therefore, from (5.1), we have

$$\sum_{(i,j)\in P} c^k_{ij} = \sum_{(p,q)\in\tilde{P}} \tilde{c}^k_{pq} \text{ for } k = 1, 2,$$

which implies that the costs of P and $\tilde{P}$ are identical. Since the vector $\tilde{c}^2$ of $\tilde{c}^2_{pq}$'s is binary, we can apply our algorithms to the network $\tilde{G}$. The time needed in this approach is polynomial in n, m and C, since the numbers of nodes and arcs in $\tilde{G}$ are linear in them.

References

1. Ahuja, R.K., Magnanti, T.L. and Orlin, J.B. (1993), *Network Flows: Theory, Algorithms and Applications*, Englewood Cliffs, NJ: Prentice Hall.
2. Avriel, M., Diewert, W.E., Schaible, S. and Zang, I. (1988), *Generalized Convexity*, New York, Plenum Press.
3. Benson, H.P. (1979), Vector maximization with two objective functions, *Journal of Optimization Theory and Applications* 28: 253–257.
4. Garey, M.R. and Johnson, D.S. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, New York: W.H. Freeman.
5. Geoffrion, M. (1967), Solving bicriterion mathematical programs, *Operations Research* 15: 39–54.
6. Guisewite, G.M. and Pardalos, P.M. (1993), A polynomial time solvable concave network flow problem, *Networks* 23: 143–149.
7. Horst, R. and Tuy, H. (1994), *Global Optimization: Deterministic Approaches*, 2nd edn., Berlin/New York: Springer Verlag.
8. Karp, R.M. and Orlin, J.B. (1981), Parametric shortest path algorithms with an application to cyclic staffing, *Discrete Applied Mathematics* 3: 37–45.
9. Klinz, B. and Tuy, H. (1993), Minimum concave-cost network flow problems with a single nonlinear arc cost, in Dungzhu Du and P.M. Pardalos (eds.), *Network Optimization Problems* (pp. 125–143), Singapore: World Scientific.

10. Konno, H. and Kuno, T. (1995), Multiplicative programming problems, in R. Horst and P.M. Pardalos (eds.), *Handbook of Global Optimization* (pp. 369–405), Dordrecht/Boston/London: Kluwer Academic Publishers.
11. Konno, H., Thach, P.T. and Tuy, H. (1997), *Optimization on Low Rank Nonconvex Structures*, Dordrecht/Boston/London: Kluwer Academic Publishers.
12. Pferschy, U. and Tuy, H. (1994), Linear programs with an additional rank two reverse convex constraint, *Journal of Global Optimization* 4: 441–454.
13. Tuy, H. (1991), Polyhedral annexation, dualization and dimension reduction technique in global optimization, *Journal of Global Optimization* 1: 229–244.
14. Tuy, H. (1992), The complementary convex structure in global optimization, *Journal of Global Optimization* 2: 21–40.
15. Tuy, H. and Al-Khayyal, F.A. (1992), Global optimization of a nonconvex single facility location problem by sequential unconstrained convex minimization, *Journal of Global Optimization* 2: 61–71.
16. Tuy, H., Dan, N.D. and Ghannadan, S. (1993), Strongly polynomial time algorithms for certain concave minimization problems on networks, *Operations Research Letters* 14: 99–109.
17. Tuy, H., Migdalas, A. and Värbrand, P. (1994), A quasiconcave minimization method for solving linear two level programs, *Journal of Global Optimization* 4: 243–264.
18. Tuy, H. and Tam, B.T. (1992), An efficient solution method for rank two quasiconcave minimization problems, *Optimization* 24: 43–56.