Research Article

The Capacity Expansion Path Problem in Networks

Jianping Li and Juanping Zhu

Department of Mathematics, Yunnan University, Kunming 650091, China

Correspondence should be addressed to Juanping Zhu; juanpingzhu@gmail.com

This paper considers the general capacity expansion path problem (GCEP) for the telecommunication operators. We investigate the polynomial equivalence between the GCEP problem and the constrained shortest path problem (CSP) and present a pseudopolynomial algorithm for the GCEP problem, no matter the graph is acyclic or not. Furthermore, we investigate two special versions of the GCEP problem. For the minimum number arc capacity expansion path problem (MN-CEP), we give a strongly polynomial algorithm based on the dynamic programming. For the minimum-cost capacity expansion shortest path problem (MCESP), we give a strongly polynomial algorithm by constructing a shortest paths network.

1. Introduction

Currently Chinese telecommunication operators are under the huge pressure of communication capacity and communication quality, followed by the rapidly increasing users. Shortest path problem plays an important role in the telecommunication networks. In this paper, we consider the network capacity expansion strategy for the telecommunication operators, called the general capacity expansion path problem (GCEP), given a weighted digraph $D = (V, A; w, c, p)$ with n nodes, m arcs, and two fixed nodes s, t. For each arc $e \in A$, nonnegative $w(e)$ is the weight, $c(e)$ is the capacity, and nonnegative $p(e)$ is the cost of expanding the capacity of the arc e by one unit. Given two positive W and d, find a path $P_{s,t}$ from s to t to satisfy two constraints: (1) $w(P_{s,t}) = \sum_{e \in P_{s,t}} w(e) \leq W$ and (2) for some arc $e \in A$, if the capacity $c(e) < d$, we must expand the capacity of arc e to d. The objective is to minimize the total expansion cost $\sum_{e \in P_{s,t}} \text{add}(e)p(e)$ on the path $P_{s,t}$, where $\text{add}(e) = d - c(e)$ if $c(e) < d$ and $\text{add}(e) = 0$ otherwise. In telecommunication network, W is the given maximum length of the path $P_{s,t}$, and d is the expected maximum number of the users. If the length of the path $P_{s,t}$ exceeds W, the communication quality would drop quickly. If the capacity (or bandwidth) of the arc is not wide enough, the arc would be congested, and we have to expand this arc to the expected d.

Krumke et al. [1] studied budget constrained network upgrading problems. The goal is how to find a minimum cost set of vertices to be upgraded so that the resulting network has a minimum spanning tree of weight no more than the budget. Their upgrading costs are on the vertices.

Zhang et al. [2] studied the problem how to increase the capacities of the elements in an edge set E efficiently so that the capacity of a given family $\mathscr{F}$ of subsets of E can be increased to the maximum extent, where the total cost for the increment of capacity is within a given budget bound D. Yang and Zhang [3] considered a class of bottleneck capacity expansion problems, which aimed to enhance bottleneck capacity to a certain level with minimum cost. There were two types of expanding models: arc expanding and node expanding. They emphasized the maximum capacity path problem. Our models are different from theirs. For example, when we consider the arc expansion cost of the path in the telecommunication network, on which each signal is sent from the source s via some transferring facility v_i to the sink t, we must make sure that the transferring length does not exceed the length bound W.

Different from our model and method, Seref et al. [4] studied an incremental optimization problem. They started from a feasible solution x^0 and made an incremental change in x^0 that would result in the greatest improvement in the objective function. For the incremental shortest path

problem, they showed that the arc inclusion version is polynomially solvable whereas the arc exclusion version is *NP*-complete.

In this paper, we present a proof of polynomial equivalence between the GCEP problem and the constrained shortest path problem (CSP), which leads to that the GCEP problem remains *NP*-hard. We design a pseudopolynomial algorithm and a FPTAS to solve the GCEP problem. Then we study two special versions of the GCEP problem. For the first version, minimum number arc capacity expansion path problem (MNCEP), we present a strongly polynomial algorithm based on dynamic programming. For the second version, the minimum-cost capacity expansion shortest path problem (MCESP), we construct a shortest path network and present a strongly polynomial algorithm.

2. The Equivalence between the GCEP Problem and the CSP Problem

Motivated by the polynomial equivalence between the Hitchcock problem and the minimum-cost flow problem [5], we prove that the GCEP problem is polynomially equivalent to the constrained shortest path problem, which leads to Theorem 1. Hassin [6] studied the constrained (or restricted) shortest path problem (CSP) as follows. Let $D = (V; A)$ be a weighted digraph with n nodes, m arcs, and two fixed nodes s, t. Each arc $e \in A$ has a length $c(e)$ and a transition time $w(e)$. These numbers are assumed to be positive integers. For any path $P_{s,t}$ from s to t in D, the length $c(P_{s,t})$ and transition time $w(P_{s,t})$ are defined as the sum of the lengths and transition times of all arcs on $P_{s,t}$, respectively. The CSP problem is to find the minimum length path $P_{s,t}$ in D such that transition time $w(P_{s,t})$ along this path does not exceed a given bound W. For convenience, we denote a notation, by $\{D = (V, A; w, c; s, t); W\}_{\text{CSP}}$, to represent an instance of the CSP problem. The CSP problem is a classical *NP*-hard problem [7]. To obtain Theorem 1, we use $\{D = (V, A; w, c, p; s, t); W; d\}_{\text{GCEP}}$ to represent an instance of the GCEP problem.

Theorem 1. *The GCEP problem is polynomially equivalent to the CSP problem.*

Proof. It is sufficient to prove that the GCEP problem can be transformed to the CSP problem in polynomial operations, and vice versa.

For any instance $\mathcal{I}$, say $\{D = (V, A; w, c, p; s, t); W; d\}_{\text{GCEP}}$, of the GCEP problem, we construct an instance $\tau(\mathcal{I})$ of the CSP problem: a digraph $\{D' = (V, A; w', c'; s, t); W'\}_{\text{CSP}}$ consisting of the same structure as the digraph D with n nodes, m arcs, two fixed nodes s and t, and two positive d and $W'(= W)$. For each arc $e \in A$, define two nonnegative $w'(e) = w(e)$ and $c'(e) = \text{add}(e)p(e)$, where $\text{add}(e) = d - c(e)$ if $c(e) < d$ and $\text{add}(e) = 0$ otherwise. The objective is to find a path $P_{s,t}$ from s to t in D' such that $\sum_{e \in P_{s,t}} w'(e) \leq W'$ and the value of $\sum_{e \in P_{s,t}} c'(e)$ is minimized.

On the converse direction, for any instance $\mathcal{I}$, say $\{D' = (V, A; w', c'; s, t); W'\}_{\text{CSP}}$, of the CSP problem, we construct an instance $\alpha(\mathcal{I})$ of the GCEP problem: a digraph $D = (V, A; w, c, p; s, t; W, d)_{\text{GCEP}}$ consisting of the same structure as the digraph D' with n nodes, m arcs, a fixed positive integer $d = 2$, and the bound $W = W'$. For each arc $e \in A$, define nonnegative $w(e) = w'(e)$, $c(e) = 1$, and $p(e) = c'(e)$, which implies that $\text{add}(e) = 1$ and $\text{add}(e)p(e) = c'(e)$. The objective is to find a path $P_{s,t}$ from s to t in D' such that $\sum_{e \in P_{s,t}} w(e) \leq W(= W')$ and the value of $\sum_{e \in P_{s,t}} \text{add}(e)p(e) = \sum_{e \in P_{s,t}} c'(e)$ is minimized.

It is easy to prove the claim: there is an optimal solution to the instance $\mathcal{I}$ of the CSP problem with the optimal value k if and only if there is an optimal solution to the instance $\alpha(\mathcal{I})$ of the GCEP problem with the optimal value k. Moreover, the transformation is executed in polynomial operations. □

Since the CSP problem is *NP*-hard, the proof of Theorem 1 implies that the GCEP problem is *NP*-hard, even if $c(e) = 1$. And if $p(e) = 1$, the GCEP problem is reduced to regular CSP problem.

3. A Pseudopolynomial Algorithm for the GCEP Problem

According to Theorem 1, for any instance $\mathcal{I}$ of the GCEP problem in the acyclic digraphs, by constructing an instance $\tau(\mathcal{I})$ of the CSP problem and utilizing the FPTAS in [6, 8] for the CSP problem, we can get a FPTAS to solve the GCEP problem. Now for the GCEP problem, no matter acyclic digraphs or not, by applying the following dynamic programming method which is similar to the Bellman-Ford algorithm [9, 10], we give a pseudopolynomial algorithm. To describe our algorithm, for each node $u \in V$ and two integers $k \geq 0$ and $g \geq 0$, define $f_k(u, g)$ to be the minimum length of an optimal path $P_{s,u}$ from s to u with respect to the weight w, such that at most k arcs are traversed and the expansion cost along $P_{s,u}$ is at most g. Set $f_k(u, g) := +\infty$ if no such path exists. For convenience, denote $\text{addcost}(e) = \text{add}(e)p(e)$ as the expansion cost for each arc $e \in A$ and W-path as the directed path if its weight is less than or equal to W. Set OPT to be the minimum capacity expansion cost of s-t W-path.

Algorithm 2 (EXACT(W)). *Comment.* Return minimum capacity expansion path $P_{s,t}$ from s to t with constraint $w(P_{s,t}) \leq W$.

Begin

Step 1 (initialization). Set $f_0(s, g) = 0$ for each $g \geq 0$ and set $f_0(u, g) = +\infty$ if $u \neq s$.

Step 2

For $g = 1, 2, \ldots, \text{OPT}$, do

For $k = 1, 2, \ldots, n - 1$, do

For each $u \in V$, do

$$f_k(u, g) = \min\{f_{k-1}(u, g), \min_{\{v|(v,u)\in A, \text{addcost}(v,u)\leq g\}} \{f_{k-1}(v, g - \text{addcost}(v, u)) + w(v, u)\}\};$$

If $f_{n-1}(t, g) \leq W$, output OPT $= g$ and the path $P_{s,t}$, and exit.

End of Algorithm EXACT(W).

Actually algorithm EXACT(W) is a dynamic programming algorithm, which generalizes the Bellman-Ford method [9, 10]. Note that OPT is not known but it satisfies OPT $= \min\{g \mid f_{n-1}(t, g) \leq W\}$. k is the maximum number of traversing arcs on the path from node s to u, and the algorithm enumerates each $k(\leq n-1)$ and each $u \in V$ for $f_k(u, g)$. To get $f_{n-1}(t, g)$, the algorithm computes $f_k(u, g)$ first for $g = 1, k = 1, \ldots, n-1$, and each $u \in V$, then for $g = 2, k = 1, \ldots, n-1$, and each $u \in V$ and so on, until the first value of g for which $f_{n-1}(t, g) \leq W$. According to the definition of $f_k(u, g)$, $f_k(t, g)$ is the minimum length of an optimal path $P_{s,t}$ from s to t with respect to the weight w, such that at most k arcs are traversed and the expansion cost along $P_{s,t}$ is at most g. $f_{n-1}(t, g) \leq W$ indicates the length with respect to weight w on this optimal path $P_{s,t}$ is less than and equal to W (optimal condition), and $g \leq$ OPT. Hence OPT is set to this value of g. So the complexity of algorithm EXACT(W) is $\mathcal{O}(\text{OPT} \cdot mn^2)$, and we obtain Theorem 3.

Theorem 3. *For the GCEP problem, the algorithm EXACT(W) produces an optimal solution, and its time complexity is $\mathcal{O}(\text{OPT} \cdot mn^2)$.*

Moreover, the technique of rounding and scaling for acyclic digraphs in [8] is applied to our pseudopolynomial algorithm EXACT(W), and we can present an FPTAS in time $\mathcal{O}(n^2 m/\varepsilon)$ for the GCEP problem even in general digraphs, where $\varepsilon > 0$. The construction of our FPTAS for general digraphs is similar to the one due to [8] for acyclic digraphs. The reader can find the algorithm in details from [8]; thus we omit it here.

4. Dynamic Programming Algorithm for the MNCEP Problem

In this section, we study the first special version of the GCEP problem, minimum number arc capacity expansion path problem (MNCEP), where for each arc $e \in A$, $p(e) = 1$, $\text{add}(e) = 1$ if $c(e) < d$ and $\text{add}(e) = 0$ otherwise. Equivalently, the objective is to find a path $P_{s,t}$ of D, on which the number of arcs to be expanded is minimized.

Like the GCEP problem, no matter acyclic digraphs or not, by applying the method similar to Bellman-Ford algorithm [9, 10], we can design a strongly polynomial algorithm to solve the MNCEP problem. For each node $u \in V$, given two integers $i \geq 0$ and $j \geq 0$, define $f(u, i, j)$ to be the minimum length of the path $P_{s,u}$ from s to u with respect to the weight w, which traverses at most i arcs with exact j arcs to be expanded. Set $f(u, i, j) := +\infty$ if no such path exists.

Algorithm 4 (MNCEP). *Comment.* Return the minimum number of arcs to be expanded on the path $P_{s,t}$ from s to t with constraint $w(P_{s,t}) \leq W$.

Begin

Step 1 (initialization). Set $f(s, i, j) = 0$ and set $f(u, i, j) = +\infty$ if $u \neq s$.

Step 2

For $i = 1, 2, \ldots, n-1$, do

For $j = 1, 2, \ldots, i$, do

For each node $u \in V$, do $f(u, i, j) = \min\{f(u, i-1, j), \min_{\{v|(v,u)\in A, c(v,u)\geq d\}}\{f(v, i-1, j) + w(v, u)\}, \min_{\{v|(v,u)\in A, c(v,u)<d\}}\{d(v, i-1, j-1) + w(v, u)\}\}$

If $f(t, i, j) \leq W$, then output OPT $= j$ and the path $P_{s,t}$, and exit.

End of Algorithm MNCEP.

The algorithm MNCEP is also based on dynamic programming. Different from algorithm EXACT(W), no unknown OPT appears in the algorithm. i is the maximum number of traversing arcs on the path from node s to u, and j is the exact number of arcs to be expanded on this path. So $j \leq i$. The algorithm enumerates each $i(\leq n-1)$, each $j \leq i$, and each $u \in V$ till $f(t, i, j) \leq W$. So the algorithm computes $f(u, i, j)$ first for $i = 1, j = 1, 2, \ldots, i$, and each $u \in V$, then for $i = 2, j = 1, 2, \ldots, i$, and each $u \in V$, and so on, until the first value $i \leq n-1$ for which $f(t, i, j) \leq W$. Hence, output OPT $= j$. Thus, the algorithm runs in $\mathcal{O}(mn^2)$ time. We obtain Theorem 5.

Theorem 5. *For the MNCEP problem, the algorithm MNCEP produces an optimal solution, and its time complexity is $\mathcal{O}(mn^2)$.*

5. Strongly Polynomial Algorithm for the MCESP Problem

The second special version of GCEP is the minimum-cost capacity expansion shortest path problem (MCESP). In MCESP problem, the bound W is set to be the shortest distance $\text{dist}(s, t)$ from s to t in D, with respect to weight w. So the objective of MCESP problem is to find a shortest path $P_{s,t}$ from s to t with respect to the weight w, such that the total expansion cost $\sum_{e\in P_{s,t}} \text{add}(e)p(e)$ on the path $P_{s,t}$ is minimized, where $\text{add}(e) = d - c(e)$ if $c(e) < d$ and $\text{add}(e) = 0$ otherwise. To solve this MCESP problem optimally, the shortest paths network plays an important role in the algorithm.

Property 1 (see [11]). Let the vector dist represent the shortest paths distances. Then a directed path $P_{s,v}$ from the source node s to the node v is a shortest path if and only if $\text{dist}(s, v) = \text{dist}(s, u) + w(u, v)$ for every arc $(u, v) \in P_{s,v}$.

For convenience, set $\text{addcost}(e) = \text{add}(e) \cdot c(e)$ as the capacity expansion cost for each arc $e \in A$.

Algorithm 6 (MCESP). *Comment.* Return the minimum-cost capacity expansion shortest path.

Begin

Step 1. Perform the breadth-first search in the digraph D, using the arcs to satisfy the equality $\text{dist}(s, v) = \text{dist}(s, u) + w(u, v)$. Construct the shortest paths network $D' = (V, A'; w, \text{addcost})$ of D, where $A' = \{(u, v) \in A \mid \text{dist}(s, v) = \text{dist}(s, u) + w(u, v)\}$. The values of w and addcost in D' remain the same as in D.

Step 2. Find the shortest path $P_{s,t}$ from s to t in digraph D', with respect to the arc capacity expansion cost addcost. Output OPT = addcost $(P_{s,t})$ and the path $P_{s,t}$.

End of Algorithm MCESP.

Theorem 7. *The algorithm MCESP optimally solves the MCESP problem, and it runs in the time $\mathcal{O}(n^2)$.*

Proof. Property 1 implies the fact: if $P_{s,v}$ is a shortest path from the source s to some node v, then $\text{dist}(s, v) = \text{dist}(s, u) + w(u, v)$ for every arc $(u, v) \in P_{s,v}$. The converse is also true: if $\text{dist}(s, v) = \text{dist}(s, u) + w(u, v)$ for every arc (u, v) in a directed path $P_{s,v}$ from the source s to the node v, then $P_{s,v}$ must be a shortest path. The algorithm MCESP searches all m arcs for $\text{dist}(s, v) = \text{dist}(s, u) + w(u, v)$ in D. Hence, the new digraph D' includes all minimum distance path from s to t in D, with respect to the weight w. The Step 2 of the algorithm selects the minimum cost capacity expansion path $P_{s,t}$ in D'. Hence, the algorithm MCESP solves the MCESP problem optimally.

Step 1 of the algorithm needs $\mathcal{O}(n^2)$ steps to compute all $d(s, v)$ by using Dijkstra algorithm for shortest paths. Step 2 needs $\mathcal{O}(m)$ steps to perform breadth-first search. Step 2 needs $\mathcal{O}(n^2)$ steps as Step 1. Hence, the time complexity of the algorithm MCESP is $\mathcal{O}(n^2)$. □

6. Conclusion

In this paper, we focus on the general capacity expansion path problem for telecommunication operators, a new practical capacity expansion model. The polynomial equivalence between the GCEP problem and the CSP problem is proved. Then by utilizing dynamic programming technique and Bellman-Ford method [9, 10], we design a pseudopolynomial algorithm for the GCEP problem. We also present two special versions of the GCEP problem: the MNCEP problem and the MCESP problem. For the MNCEP problem, by using dynamic programming method and Bellman-Ford method, we design a strongly polynomial algorithm to solve it. For the MCESP problem, by constructing a shortest paths network, we design a strongly polynomial algorithm.

For further work, we would investigate some other special versions of the GCEP problem and related algorithms, which exist in the current telecommunication networks.

Acknowledgment

References

[1] S. O. Krumke, M. V. Marathe, H. Noltemeier et al., "Improving minimum cost spanning trees by upgrading nodes," *Journal of Algorithms*, vol. 33, no. 1, pp. 92–111, 1999.

[2] J. Zhang, C. Yang, and Y. X. Lin, "A class of bottleneck expansion problems," *Computers and Operations Research*, vol. 28, no. 6, pp. 505–519, 2001.

[3] C. Yang and J. Z. Zhang, "On the bottleneck capacity expansion problem on networks," *Acta Mathematica Scientia*, vol. 26, no. 2, pp. 202–208, 2006.

[4] O. Seref, R. K. Ahuja, and J. B. Orlin, "Incremental network optimization: theory and algorithms," *Operations Research*, vol. 57, pp. 586–594, 2009.

[5] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prince Hall, 1982.

[6] R. Hassin, "Approximation schemes for the restricted shortest path problem," *Mathematics of Operations Research*, vol. 17, no. 1, pp. 36–42, 1992.

[7] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, Calif, USA, 1979.

[8] F. Ergun, R. Sinha, and L. Zhang, "An improved FPTAS for restricted shortest path," *Information Processing Letters*, vol. 83, no. 5, pp. 287–291, 2002.

[9] R. Bellman, "On a routing problem," *Quarterly of Applied Mathematics*, vol. 16, pp. 87–90, 1958.

[10] L. R. Ford, "Network flow theory," *The RAND Corporation Paper*, p. 923, 1956.

[11] R. K. Ahuja, L. Magnanti, and B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Printice Hall, 1993.