Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 394721, 8 pages
doi:10.1155/2012/394721

Research Article

Polynomial Time Approximation Schemes for the Constrained Minimum Spanning Tree Problem

Yen Hung Chen

Department of Computer Science, Taipei Municipal University of Education, No. 1, Ai-Guo West Road, Taipei 10048, Taiwan

Correspondence should be addressed to Yen Hung Chen, yhchen@tmue.edu.tw

Received 28 October 2011; Accepted 18 January 2012

Academic Editor: Rudong Chen

Let $G = (V, E)$ be an undirected graph with a weight function and a cost function on edges. The constrained minimum spanning tree problem is to find a minimum cost spanning tree T in G such that the total weight in T is at most a given bound B. In this paper, we present two polynomial time approximation schemes (PTASs) for the constrained minimum spanning tree problem.

1. Introduction

Motivated by the applications of quality of service (QoS) routing and multicasting, several multiple criteria problems have been studied (see [1–6] and references therein). Marathe et al. [5] studied a class of *bicriteria network design* problems that are defined as follows. Given an undirected graph $G = (V, E)$ and two independent minimization criteria with a bound on the first criterion, a *generic bicriteria network design* problem involves the minimization of the second criterion but satisfies the bound on the first criterion among all possible subgraphs from G [5]. They considered three different criteria that have total edge cost, diameter, and maximum degree of a graph. The total edge cost of a graph is the sum of the costs of all edges in the subgraph. The diameter of a graph is the maximum distance between any pair of nodes in the subgraph. The degree of a node is the number of nodes adjacent to this node, and the maximum degree of a graph is maximum over the degrees of all nodes in the subgraph. Let $G = (V, E)$ be an undirected graph and a positive integer B. Two different nonnegative functions, *weight* and *cost,* associate on edges in E, respectively. The *constrained minimum spanning tree* (CMST) problem [7–10] is to find a minimum total cost spanning tree T in G such that the total weight in T is at most B. Aggarwal et al. [7] showed that the CMST problem is weakly NP-hard. Then Hong et al. [9] gave a pseudopolynomial time algorithm to solve the CMST problem. Moreover, some approximation algorithms have been proposed to solve the CMST problem [5, 8, 10].

Given two positive real numbers α and β, an (α, β)-approximation algorithm for the CMST problem is defined as a polynomial-time algorithm that produces a solution with the total weight at most α times the bound B and the total cost at most β times the total cost of optimal solution for the CMST problem. The approximation ratio is denoted by (α, β). An (α, β)-approximation algorithm for the CMST problem is called the polynomial time approximation scheme (PTAS) that produces a spanning tree with an approximation ratio of $(1, 1+\varepsilon)$, for any constant $\varepsilon > 0$. For any constant $\varepsilon > 0$, Marathe et al. [5] gave a $(1+\varepsilon, 1+1/\varepsilon)$-approximation algorithm for the CMST problem. Ravi and Goemans [10] presented a $(2, 1)$-approximation algorithm to solve the CMST problem in $O(|E|\log^2|V| + |V|\log^3|V|)$ time and improved the ratio to $(1+\varepsilon, 1)$, where the time complexity is $O(|V|^{O(1/\varepsilon)}(|E|\log^2|V| + |V|\log^3|V|))$ for any constant $\varepsilon > 0$. These algorithms are based on the Lagrangian relaxation. Hassin and Levin [8] employed Ravi and Goemans' [10] algorithm with the matroid intersection strategy [11–13] to design a $(1, 1+5\varepsilon)$-approximation algorithm in $O((1/\varepsilon^2)^{(1/\varepsilon)}|V|^3)$ time for the CMST problem. For other related bicriteria problems, Hassin [14] gave a fully polynomial time approximation scheme (FPTAS) for the *restricted shortest path* problem. The purpose of this problem is to find a minimum cost shortest path P from a source to a destination subject to the constraint that the total weight in P is at most B. Then, Lorenz and Raz [15] gave an improved FPTAS whose time complexity is $O(|V||E|\log\log|V| + (|V||E|)/\varepsilon)$ for the restricted shortest path problem. Further, Xue et al. [16] also designed an FPTAS with improved time complexity to $O(|V||E|\log\log\log|V| + (|V||E|)/\varepsilon)$ for the restricted shortest path problem. If the graph is directed acyclic, and Ergun et al. [17] provided an improved FPTAS for the restricted shortest path problem with running time $O((|V||E|)/\varepsilon)$. Chen and Xue [18, 19] used a similar method to design PTASs for the *k-pair delay constrained minimum cost routing* problem and the *weight constrained Steiner tree* problem in series-parallel graphs. These algorithms are based on the rounding and scaling strategy that is offered by Hassin [14]. In this paper, we present two polynomial time approximation schemes to find $(1, 1+\varepsilon)$-approximation ratio for the CMST problem. Applying the second PTAS to Hassin and Levin's algorithm [8], the approximation ratio can be improved to $(1, 1+4\varepsilon)$ for the CMST problem.

The rest of this paper is organized as follows. In Section 2, we first describe a $(1, (1+\varepsilon)^2)$-approximation algorithm for the CMST problem and then improve the approximation ratio to $(1, 1+\varepsilon)$. Finally, we make a conclusion in Section 3.

2. PTASs for the Constrained Minimum Spanning Tree Problem

In this section, we first clarify a $(1, (1+\varepsilon)^2)$-approximation algorithm for the CMST problem. Hassin and Levin [8] first used this algorithm to obtain a $(1, 2(1+\varepsilon))$ ratio for the CMST problem. Then they applied the Lagrangian relaxation and matroid intersection methods to improve the ratio to $(1, 1+5\varepsilon)$ in $O((1/\varepsilon^2)^{(1/\varepsilon)}|V|^3))$ time. However, they did not describe the details of this algorithm. Then, we improve this approximation ratio into $(1, 1+\varepsilon)$. Note that applying our algorithm, the approximation ratio of Hassin and Levin's algorithm [8] can be improved to $(1, 1+4\varepsilon)$ ratio with the same time complexity. For convenience, we use $C(H)$ (resp., $W(H)$) to denote the sum of the costs (resp., weights) of all edges in any subgraph H of G. Let T_{OPT} be the optimal solution of the CMST problem. For any constant $\varepsilon > 0$ and an integer B, Ravi and Goemans' [10] algorithm, denoted by $\mathcal{A}_R(\varepsilon, B)$, produces a spanning tree

T of the $W(T)$ at most $(1+\varepsilon)B$ and of the $C(T)$ at most $C(T_{\text{OPT}})$. Our algorithm first finds a cost of the lower bound, denoted by LB, and a cost of the upper bound, denoted by UB, for the CMST problem. Then, iteratively shrink this range between LB and UB until UB $\leq (1+\varepsilon)$LB. A trivial lower bound LB can be set to 1, and an upper bound UB can be set to $|V| * C_{\max}$, in which $C_{\max}$ is the maximum edge cost in G. However, the range between the trivial lower bound and upper bound is too large (i.e., not polynomial in input size). We reduce the range by Lorenz and Raz's method [15] such that UB $\leq |V| *$ LB. First, let $c_1, c_2, \ldots, c_k$ be all the distinct costs of edges in G. $G_j = (V, E_j)$ denotes the subgraph of $G(V, E)$, where E_j is the set of edges with costs not greater than c_j. It is clear that $G_k = G$ and $G_j \subseteq G_{j+1}$ for $1 \leq j \leq k-1$. For $1 \leq j \leq k$, we can use Prim's algorithm [20] to find a minimum weight spanning tree T_j in G_j. Let J be the smallest index j such that $W(T_j)$ is no more than B. Then, finding the value c_J of a graph G can be done in $O(|V|^2 \log |V|)$ time by binary search and Prim's algorithm [20]. Therefore, we have $c_J \leq C(T_{\text{OPT}}) \leq |V| * c_J$. Hence, we can let the initial UB be $|V| * c_J$ and LB be c_J. If UB $\leq (1+\varepsilon)$LB, then $C(T_J) \leq$ UB $\leq (1+\varepsilon)$LB $\leq (1+\varepsilon)C(T_{\text{OPT}})$. Otherwise, we first swap the cost (resp., weight) to the weight (resp., cost) for each edge in G and find a value M between LB and UB. Then, apply algorithm $\mathcal{A}_R(\varepsilon, M)$ to find a spanning tree T in G. If the total weight of T is no more than B, let UB $= (1+\varepsilon)M$; otherwise, let LB $= M$. Hence, the range between LB and UB is shrunk. Then, we repeatedly run the algorithm $\mathcal{A}_R(\varepsilon, M)$ until UB $\leq (1+\varepsilon)$LB. The next lemma shows that we can obtain either a lower bound M or an upper bound $(1+\varepsilon)M$ for the CMST problem after running $\mathcal{A}_R(\varepsilon, M)$.

Lemma 2.1. *Given an instance of the CMST problem with a constant $\varepsilon > 0$ and an integer $\mathcal{C}$, one swaps the cost (resp., weight) to the weight (resp., cost) for each edge and then applies algorithm $\mathcal{A}_R(\varepsilon, \mathcal{C})$ to find a spanning tree T. If $W(T) \leq B$, then there is an upper bound $(1+\varepsilon)\mathcal{C}$. Otherwise, one can obtain a lower bound $\mathcal{C}$.*

Proof. For convenience, we use $T_{\mathcal{C}}$ to denote the minimum weight spanning tree of G with total cost at most $\mathcal{C}$. After running the algorithm $\mathcal{A}_R(\varepsilon, \mathcal{C})$, we have a spanning tree T with $W(T) \leq W(T_{\mathcal{C}})$ and $\mathcal{C}(T) \leq (1+\varepsilon)\mathcal{C}$ since the costs and weights of all edges are exchanged. Hence, if $W(T) \leq B$, then $\mathcal{C}(T_{\text{OPT}}) \leq \mathcal{C}(T) \leq (1+\varepsilon)\mathcal{C}$. Otherwise, $W(T_{\mathcal{C}}) \geq W(T) > B$. Then we assume $\mathcal{C}(T_{\text{OPT}}) < C$ for contradiction. Hence, we have $W(T_{\mathcal{C}}) \leq W(T_{\text{OPT}})$. However, $W(T_{\text{OPT}}) \leq B$, and we have $W(T_{\mathcal{C}}) \leq B$. Hence, $\mathcal{C}$ must be less than or equal to $C(T_{\text{OPT}})$. □

For clarification, we describe the $(1, (1+\varepsilon)^2)$-approximation algorithm for the CMST problem as follows.

Algorithm PTAS-CMST

Input: A graph $G = (V, E)$ with a weight function and a cost function on edges, an integer bound B, and two real numbers $\varepsilon > 0$ and $\rho > 0$.
Output: A spanning tree T_{APX} of G subject to $W(T_{\text{APX}}) \leq B$.

```
  (1) For each edge in E, swap cost to weight and weight to cost, respectively. end for
  (2) Let an initial upper bound UB = $|V| * c_J$ and an initial lower bound LB = $c_J$.
  (3) If UB $\leq (1+\varepsilon)$LB, then let $T_{APX} = T_J$ and return.

      Else
(3.1) Let $L$ = LB and $U = \lceil UB/(1+\rho) \rceil$.
```

(3.2) Repeat the following steps until the condition (1) $U \leq (1+\varepsilon)L$ or (2) $U \leq L+1$ holds.

(3.2.1) Let a middle value $M = \sqrt{U * L}$.

(3.2.2) Use algorithm $\mathscr{A}_R(\rho, M)$ to find a spanning tree T.

(3.2.3) /* **if** $W(T) > B$, **then** we have a lower bound M; **else** we have an upper bound $(1+\rho)M$ and $C(T)$ is less than or equal to this upper bound. */

If $W(T) \leq B$, **then** let $U = M$ and $T_{\text{APX}} = T$; **else** let $L = M$.

The approximation ratio of Algorithm PTAS-CMST is shown in the next lemma.

Lemma 2.2. *Algorithm PTAS-CMST returns a spanning tree T_{APX} with $C(T_{APX}) \leq (1+\varepsilon)(1+\rho)C(T_{OPT})$.*

Proof. Let the upper bound U_i, the middle value M_i, and the lower bound L_i be parameters at the beginning of the ith iteration of step (3.2) for Algorithm PTAS-CMST. Let T_i denote a spanning tree after running the i-th iteration of step (3.2.2). It is clear that $L_i \leq M_i \leq U_i$. Hence, $U_{i+1} \leq U_i$ and $L_{i+1} \geq L_i$. Moreover, we have $L_{i+1} = \sqrt{U_i * L_i}$ and $U_{i+1} = U_i$ when $W(T_i) > B$. Then, we also have $U_{i+1} = \sqrt{U_i * L_i}$ and $L_{i+1} = L_i$ when $W(T_i) \leq B$. Further, $U_{i+1}/L_{i+1} = U_i/\sqrt{U_i * L_i}$, or $U_{i+1}/L_{i+1} = \sqrt{U_i * L_i}/L_i$. Therefore, we have $U_{i+1}/L_{i+1} = \sqrt{U_i/L_i}$. Note that it successively narrows down the range between the lower bound L and the upper bound U after each iteration of step (3.2). After running the ith iteration of step (3.2.3), we obtain either a lower bound M_i or an upper bound $(1+\rho)M_i$ by Lemma 2.1. Let k be the number of iterations of step (3.2). After performing this algorithm, we have either $U_{k+1} \leq (1+\varepsilon)L_{k+1}$ or $U_{k+1} \leq L_{k+1}+1$. For the former case, we have

$$\begin{aligned} C(T_{\text{APX}}) &\leq (1+\rho)U_{k+1} \leq (1+\varepsilon)(1+\rho)L_{k+1} \\ &\leq (1+\varepsilon)(1+\rho)C(T_{\text{OPT}}). \end{aligned} \tag{2.1}$$

For the latter case, we have

$$\begin{aligned} C(T_{\text{APX}}) &\leq (1+\rho)U_{k+1} \leq (1+\rho)(L_{k+1}+1) \\ &\leq (1+\rho)(C(T_{\text{OPT}})+1). \end{aligned} \tag{2.2}$$

Let ρ be ε, and hence Algorithm PTAS-CMST achieves an approximation ratio of $(1,(1+\varepsilon)^2)$ for the CMST problem. □

Therefore, we have the following theorem.

Theorem 2.3. *The CMST problem admits a PTAS. For any constant $\varepsilon > 0$, a $(1,(1+\varepsilon)^2)$-approximation algorithm for the CMST problem can be found in polynomial time.*

Proof. By Lemma 2.2, Algorithm PTAS-CMST returns a $(1,(1+\varepsilon)^2)$-approximation solution for the CMST problem. Next, we analyze the time complexity of Algorithm PTAS-CMST as follows. Let k be the number of iterations of step (3.2). We also let L_1 and U_1 be the initial lower bound and upper bound, respectively. Since $U_{i+1}/L_{i+1} = \sqrt{U_i/L_i}$, we have

$$
\begin{gathered}
\frac{U_1}{L_1}, \\
\left(\frac{U_2}{L_2}\right) = \left(\frac{U_1}{L_1}\right)^{1/2}, \\
\left(\frac{U_3}{L_3}\right) = \left(\frac{U_2}{L_2}\right)^{1/2} = \left(\frac{U_1}{L_1}\right)^{(1/2)^2}, \\
\vdots \\
\left(\frac{U_k}{L_k}\right) = \left(\frac{U_{k-1}}{L_{k-1}}\right)^{1/2} = \cdots = \left(\frac{U_1}{L_1}\right)^{((1/2)^{k-1})}.
\end{gathered}
\tag{2.3}
$$

Because $(U_k/L_k) > (1+\varepsilon)$ and $U_1/L_1 \leq (|V|/1+\rho)+1$, we have $k \leq \log_{1/2}\log_{(|V|+1+\rho)/(1+\rho)}(1+\varepsilon)+1$. Hence, k is $O(\log\log|V| - \log\log(1+\varepsilon))$. Then, step (3.2.2) can be done in $O(|V|^{o(1/\rho)} * (|E|\log^2|V| + |V|\log^3|V|))$ time by Ravi's algorithm [10]. Hence, Algorithm PTAS-CMST is a polynomial time approximation scheme for the CMST problem. □

Now, we improve the approximation ratio to $(1, 1+\varepsilon)$ for the CMST problem. This algorithm is the modification of the one designed in [17] for the restricted shortest path problem.

Algorithm Modify-PTAS-CMST

Input: A graph $G = (V, E)$ with a weight function and a cost function on edges, an integer bound B, and two real numbers $\varepsilon > 0$ and $0 < \gamma < 1$.
Output: A spanning tree T_{APX} of G subject to $W(T_{\mathrm{APX}}) \leq B$.

```
  (1) For each edge in $E$, swap cost to weight and weight to cost, respectively. end for
  (2) Let an initial upper bound UB = $|V| * c_J$ and an initial lower bound LB = $c_J$.
  (3) IfUB $\leq (1+\varepsilon)$LB,then let $T_{\mathrm{APX}} = T_J$ and return.
      Else
(3.1) Let $L$ = LB and $U$ = UB.
(3.2) Repeat the following steps until the condition (1) $U \leq (1+\varepsilon)L$ or (2) $U \leq L+1$
      holds.
    (3.2.1) Let a real number $\rho = (U/L)^{\gamma} - 1$ and a middle value $M = \sqrt{U * L/(1+\rho)}$.
    (3.2.2) Use algorithm $\mathrm{A}_R(\rho, M)$ to find a spanning tree $T$.
    (3.2.3) /* if $W(T) > B$, then we have a lower bound $M$; else we have an upper bound
            $(1+\rho)M$ and $C(T)$ is less than or equal to this upper bound. */
      If $W(T) \leq B$, then let $U = (1+\rho)M$ and $T_{\mathrm{APX}} = T$; else let $L = M$.
```

The approximation ratio of Algorithm Modify-PTAS-CMST is shown in the next lemma.

Lemma 2.4. *The spanning tree T_{APX} is a $(1, 1+\varepsilon)$-approximation solution for the CMST problem.*

Proof. Let the upper bound U_i, the middle value M_i, the lower bound L_i, and the real number ρ_i be parameters at the beginning of the ith iteration of step (3.2) for Algorithm Modify-PTAS-CMST. Let T_i denote a spanning tree after running the i-th iteration of step (3.2.2). If $W(T_i) \leq B$, we have $U_{i+1} = (1+\rho_i)M_i = (U_i^{1+\gamma} * L_i^{1-\gamma})^{1/2}$ and $L_{i+1} = L_i$. Otherwise, if $W(T_i) > B$, we have $L_{i+1} = M_i = (U_i^{1-\gamma} * L_i^{1+\gamma})^{1/2}$ and $U_{i+1} = U_i$. Hence, $(U_{i+1}/L_{i+1}) = (U_i/L_i)^{((\gamma+1)/2)}$. Note that it successively narrows down the range between the lower bound L and the upper bound U after each iteration of step (3.2) since $0 < \gamma < 1$. Let k be the number of iterations of step (3.2). After performing this algorithm, we have either $U_{k+1} \leq (1+\varepsilon)L_{k+1}$ or $U_{k+1} \leq L_{k+1} + 1$. For the former case, we have

$$\begin{aligned} C(T_{\mathrm{APX}}) &\leq U_{k+1} \leq (1+\varepsilon)L_{k+1} \\ &\leq (1+\varepsilon)C(T_{\mathrm{OPT}}). \end{aligned} \tag{2.4}$$

For the latter case, we have

$$C(T_{\mathrm{APX}}) \leq U_{k+1} \leq (C(T_{\mathrm{OPT}}) + 1). \tag{2.5}$$

□

Therefore, we have the following theorem.

Theorem 2.5. *The CMST problem admits a PTAS. For any constant $\varepsilon > 0$, a $(1, 1+\varepsilon)$-approximation algorithm for the CMST problem can be found in polynomial time.*

Proof. Clearly, Algorithm Modify-PTAS-CMST returns a $(1, 1+\varepsilon)$-approximation solution for the CMST problem by Lemma 2.4. Next, we analyze the time complexity of Algorithm Modify-PTAS-CMST as follows. Let k be the number of iterations of step (3.2). We also let L_1 and U_1 be the initial lower bound and upper bound, respectively. Then, we have

$$\begin{gathered} \frac{U_1}{L_1}, \\ \left(\frac{U_2}{L_2}\right) = \left(\frac{U_1}{L_1}\right)^{(\gamma+1)/2}, \\ \left(\frac{U_3}{L_3}\right) = \left(\frac{U_2}{L_2}\right)^{(\gamma+1)/2} = \left(\frac{U_1}{L_1}\right)^{(((\gamma+1)/2)^2)}, \\ \vdots \\ \left(\frac{U_k}{L_k}\right) = \left(\frac{U_{k-1}}{L_{k-1}}\right)^{(\gamma+1)/2} = \cdots = \left(\frac{U_1}{L_1}\right)^{(((\gamma+1)/2)^{k-1})} \end{gathered} \tag{2.6}$$

since $U_{i+1}/L_{i+1} = (U_i/L_i)^{(\gamma+1/2)}$.

Initially, $U_1/L_1 = |V|$. After running Algorithm Modify-PTAS-CMST, we have $(U_{k+1}/L_{k+1}) \leq (1+\varepsilon)$. Step (3.2.2) can be done in $O(|V|^{o(1/\rho)} * (|E|\log^2|V| + |V|\log^3|V|))$ time by Ravi's algorithm [10]. Let I be the smallest index j such that $|V|^{o(1/\rho_j)} > |V|$, for $1 \leq j \leq k$. For all $1 \leq j \leq I - 1$, the time is $\sum_{j=1}^{I-1} O(|V|^{o(1/\rho_j)}\,(|E|\log^2|V| + |V|\log^3|V|))$. It is clear that $(U_I/L_I) < 2^{1/\gamma}$ and $(U_{I-1}/L_{I-1}) \geq 2^{1/\gamma}$. Hence, we have $\log_{(\gamma+1)/2}(1/(\gamma * \log_2|V|)) + 1 \geq I - 1 > \log_{(\gamma+1)/2}(1/(\gamma * \log_2|V|))$ since $(U_I/L_I) = (U_1/L_1)^{(((\gamma+1)/2)^{I-1})}$. If $I - 1 \geq 1$, the complexity is at most $O(|V|(1 - \log_{((\gamma+1)/2)}\gamma - \log_{(\gamma+1/2)}(\log_2|V|))(|E|\log^2|V| + |V|\log^3|V|))$, for all $1 \leq j \leq I - 1$. Then, for all $I \leq j \leq k$, the time complexity is $\sum_{j=I}^{k} O(|V|^{o(1/\rho_j)}(|E|\log^2|V| + |V|\log^3|V|))$. Since $(U_k/L_k) > (1+\varepsilon)$ with $(U_k/L_k) = (U_1/L_1)^{(((\gamma+1)/2)^{(\mathrm{k}-1)})}$, we have $k < \log_{(\gamma+1)/2}(\log(1+\varepsilon)/\log|V|) + 1$. Moreover, $(1/\rho_I) \leq (1/(\rho_{I+1})) \leq \cdots \leq (1/\rho_k)$. Hence we have the time complexity at most $O(|V|^{o(1/\rho_k)}(1+\log_{(\gamma+1)/2}\log(1+\varepsilon)+\log_{(\gamma+1)/2}\gamma)(|E|\log^2|V|+|V|\log^3|V|))$ for all $I \leq j \leq k$. Because γ is a constant, the total time complexity is at most $O((|V|\log\log|V| + |V|^{1/((1+\varepsilon)^{\gamma}-1)}\log\log(1+\varepsilon))(|E|\log^2|V|+|V|\log^3|V|))$, where $1/\rho_k \leq 1/((1+\varepsilon)^{\gamma}-1)$, $0 < \gamma < 1$. □

Let $\varepsilon = 1$ and $\gamma \approx 1$. Use Algorithm Modify-PTAS-CMST to find a feasible solution for the CMST problem with ratio (1,2). The total time complexity of this algorithm is at most $O((|V|^{1/(2^{\gamma-1})} + |V|\log\log|V|)(|E|\log^2|V| + |V|\log^3|V|))$. Then, we can apply Hassin and Levin's [8] algorithm (i.e., the Lagrangian relaxation and matroid intersection), and the approximation ratio can be improved to $(1, 1+4\varepsilon)$ with the same time complexity of Hassin and Levin's algorithm (i.e., $O((1/\varepsilon^2)^{(1/\varepsilon)}|V|^3)$).

3. Conclusion

In this paper, we presented two polynomial time approximation schemes for the CMST problem. Using the second PTAS with the Lagrangian relaxation and matroid intersection, the approximation ratio of Hassin and Levin's algorithm [8] can be improved to the ratio of $(1, 1+4\varepsilon)$. A open problem left in the paper could involve studying whether there exists a fully polynomial time approximation scheme for the CMST problem.

Acknowledgment

This work was supported in part by the National Science Council of the Republic of China under Contract NSC99-2221-E-133-003.

References

[1] C. H. Chow, "On multicast path finding algorithms," in *Proceedings of the 10th Annual Joint Conference of the IEEE and Communications Societies (IEEE INFOCOM '91)*, vol. 3, pp. 1274–1283, Bal Harbour, Fla, USA, 1991.

[2] A. Goel, K. G. Ramakrishnan, D. Kataria, and D. Logothetis, "Efficient computation of delay-sensitive routes from one source to all destinations," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, pp. 854–858, Anchorage, Alaska, USA, 2001.

[3] B. Kadaba and J. Jaffe, "Routing to multiple destinations in computer networks," *IEEE Transactions on Communications*, vol. 31, pp. 343–351, 1983.

[4] V. P. Kompella, J. C. Pasquale, and G. C. Polyzos, "Multicasting for multimedia applications," *IEEE/ACM Transactions on Networking*, pp. 286–292, 1993.

[5] M. V. Marathe, R. Ravi, R. Sundaram, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt, III, "Bicriteria network design problems," *Journal of Algorithms*, vol. 28, no. 1, pp. 142–171, 1998.

[6] Z. Wang, “On the complexity of quality of service routing,” *Information Processing Letters*, vol. 69, no. 3, pp. 111–114, 1999.

[7] V. Aggarwal, Y. P. Aneja, and K. P. K. Nair, “Minimal spanning tree subject to a side constraint,” *Computers and Operations Research*, vol. 9, no. 4, pp. 287–296, 1982.

[8] R. Hassin and A. Levin, “An efficient polynomial time approximation scheme for the constrained minimum spanning tree problem using matroid intersection,” *SIAM Journal on Computing*, vol. 33, no. 2, pp. 261–268, 2004.

[9] S.-P. Hong, S.-J. Chung, and B. K. Park, “A fully polynomial bicriteria approximation scheme for the constrained spanning tree problem,” *Operations Research Letters*, vol. 32, no. 3, pp. 233–239, 2004.

[10] R. Ravi and M. Goemans, “The constrained minimum spanning tree problem,” in *Proceedings of the 5th Scandinavian Workshop on Algorithm Theory*, vol. 1097 of *Lecture Notes in Computer Science*, pp. 66–75, 1996.

[11] P. Chalasani and R. Motwani, “Approximating capacitated routing and delivery problems,” *SIAM Journal on Computing*, vol. 28, no. 6, pp. 2133–2149, 1999.

[12] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver, *Combinatorial optimization*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, New York, NY, USA, 1998, A Wiley-Interscience Publication.

[13] E. L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, New York, NY, USA, 1976.

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

[15] D. H. Lorenz and D. Raz, “A simple efficient approximation scheme for the restricted shortest path problem,” *Operations Research Letters*, vol. 28, no. 5, pp. 213–219, 2001.

[16] G. Xue, W. Zhang, J. Tang, and K. Thulasiraman, “Polynomial time approximation algorithms for multi-constrained QoS routing,” *IEEE/ACM Transactions on Networking*, vol. 16, pp. 656–669, 2008.

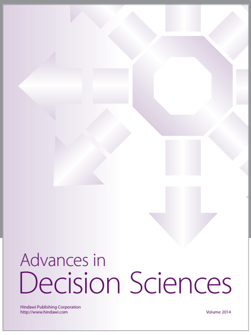
[17] F. Ergun, R. Sinha, and L. Zhang, “An improved FPTAS for restricted shortest path,” *Information Processing Letters*, vol. 83, no. 5, pp. 287–291, 2002.

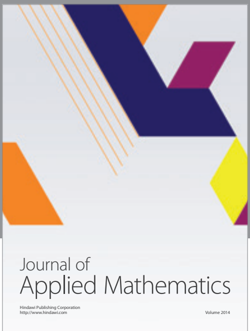
[18] G. Chen and G. Xue, “*K*-pair delay constrained minimum cost routing in undirected networks,” in *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '01)*, pp. 230–231, SIAM, Washington, DC, USA,, 2001.

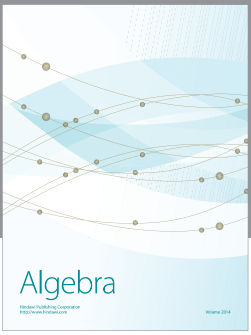
[19] G. Chen and G. Xue, “A PTAS for weight constrained Steiner trees in series-parallel graphs,” *Theoretical Computer Science*, vol. 304, no. 1–3, pp. 237–247, 2003.

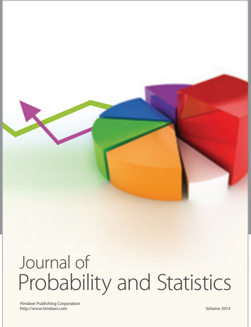
[20] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, Mass, USA, 2nd edition, 2001.

Advances in
Operations Research

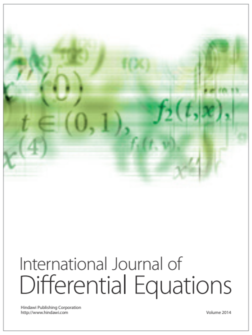

Advances in
Decision Sciences

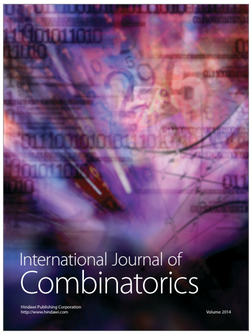

Journal of
Applied Mathematics

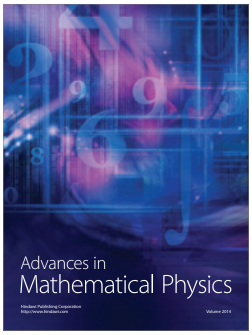

Algebra

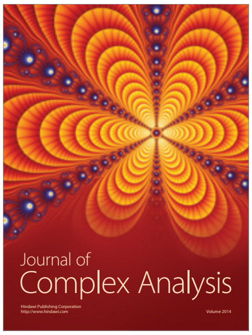

Journal of
Probability and Statistics

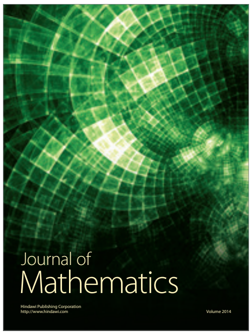
The Scientific
World Journal

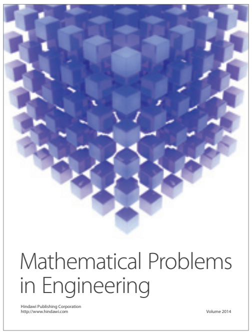

International Journal of
Differential Equations

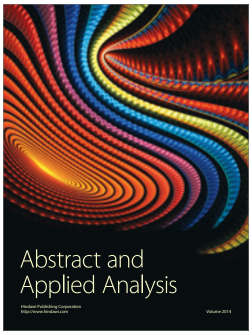

International Journal of
Combinatorics

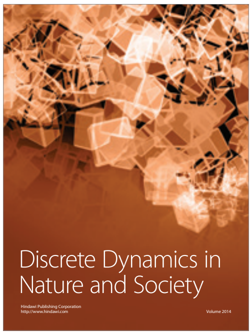

Advances in
Mathematical Physics

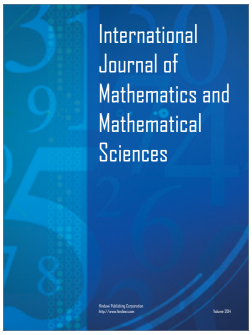

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

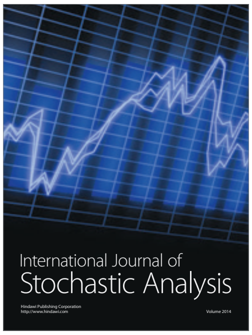

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization