Scientia Iranica E (2014) 21(3), 980–987

SCIENTIA IRANICA

Sharif University of Technology
Scientia Iranica
Transactions E: Industrial Engineering
www.scientiairanica.com

Scaling implementation of a tension rectification algorithm to solve the feasible differential problem

M. Ghiyasvand*

Department of Mathematics, Faculty of Science, Bu-Ali Sina University, Hamedan, Iran.

Received 19 May 2012; received in revised form 16 November 2012; accepted 8 June 2013

KEYWORDS
Operations research;
Network flows;
The feasible differential problem;
Tension rectification algorithm;
Scaling implementation.

Abstract. The feasible differential problem is solved using a tension rectification algorithm. In this paper, we present a scaling implementation of a tension rectification algorithm. Let n, m, U denote the number of nodes, number of arcs, and maximum arc capacity value of an arc, respectively. Our implementation runs in $O(mn \log U)$, which is $O(mn \log n)$ under the similarity assumption. The tension rectification algorithm runs in $O(m^2)$ time, so, our implementation is an improvement if $n \log n < m$. Another merit of our algorithm is that, in cases where the feasible differential problem does not have a solution, it presents some information that is useful to the modeler in estimating the maximum cost of adjusting the network.

1. Introduction

The first theoretical studies on tension were discussed by Berge and Ghouila-Houri [1,2] at the beginning of the 1960s. In 1971, Pla [3] presented an out of kilter algorithm to solve the minimum cost tension problem. Hajiat [4] showed that Pla's algorithm is not polynomial using a graph family $\{T_n, n \geq 2\}$ on which it runs necessarily in an exponential number of iterations, namely, $2^n + 2^{n-1} + 2^{n-2} - 2$ calls to a linear labeling process. Hamacher [5] developed two pseudo-polynomial time algorithms in 1985: negative cut and shortest augmenting cut algorithms. Other non-polynomial algorithms were given by Rockafellar [6]. Polynomial time algorithms to solve the minimum cost tension problem have been presented by Hadjiat and Maurras [7] and Ghiyasvand [8,9]. Piecewise linear and convex costs of the problem and inverse tension problems have been discussed in [10-14].

Let $D = (N, A)$ be a connected digraph with vertex set, N, containing n vertices, and arc set, A, containing m arcs. We denote an arc from node i to node j by (i, j). Let R^A (resp. R^N) be a collection of all ordered m-tuples (resp. n-tuples) of real numbers on set A (resp. N). A vector $\theta \in R^A$ is a *tension* on graph G with a potential $\pi \in R^N$, such that $\theta_{ij} = \pi_j - \pi_i$, for each $(i, j) \in A$. Each arc, $(i, j) \in A$, has a capacity, u_{ij}, that denotes the maximum amount of θ_{ij} on the arc and a lower bound l_{ij} that denotes the minimum amount of θ_{ij} on the arc. Tension θ is called a *feasible tension* if $l_{ij} \leq \theta_{ij} \leq u_{ij}$, for each $(i, j) \in A$. *The feasible differential problem* determines a feasible tension (if it exists).

A *cycle*, C, in a directed graph is a sequence $i_1, i_2, \ldots, i_k$ of distinct nodes of N, such that either $(i_r \to i_{r+1}) \in A$ (a *forward arc* in C) or $(i_{r+1} \to i_r) \in A$ (a *backward arc* in C) for $r = 1, 2, \ldots, k$ (where we interpret i_{k+1} as i_1). It is obvious that tension is arc-weighting, having a zero sum on every cycle of the graph, so, for each cycle, C, we have:

$$\sum_{(i,j)\in C^+} \theta_{ij} - \sum_{(i,j)\in C^-} \theta_{ij} = 0, \tag{1}$$

where C^+ and C^- are the forward and backward arcs of the cycle, respectively. For a given cycle, C, define:

*. *E-mail address: mghiyasvand@basu.ac.ir (M. Ghiyasvand)*

$$d^+(C) = \sum_{(i,j)\in C^+} u_{ij} - \sum_{(i,j)\in C^-} l_{ij}. \qquad (2)$$

The next theorem presents a necessary and sufficient condition to conclude that the feasible differential problem has a feasible tension.

Theorem 1 (Feasible differential theorem [6, page 193]). The feasible differential problem has a feasible tension if and only if $d^+(C) \geq 0$ for each cycle, C. □

The feasible differential problem can be solved using a tension rectification algorithm [6, page 203-5, and 15, page 70]. The tension rectification algorithm runs in $O(m^2)$ time ([6], page 205, line 20). When the feasible differential problem does not have a feasible tension, this algorithm only computes cycle C with $d^+(C) < 0$.

In this paper, we first present a scaling idea to solve the feasible differential problem, then a scaling implementation of the tension rectification algorithm is presented, which is a new method for solving the problem. Our algorithm runs in $O\ (mn \log U)$ time, where U is the maximum arc capacity value of an arc.

To avoid systematic underestimation of running time, in comparing two running times, sometimes, it will be assumed that the bound, U, is polynomial bounded in n, namely, $U = O(n^k)$, for some constant k [16]. This assumption is called as *the similarity assumption* [17, page 60]. Thus, under the similarity assumption [16], our algorithm runs in $O(mn \log n)$, which is an improvement if $n \log n < m$ (in comparison with the tension rectification algorithm [6,15]).

In cases where the feasible differential problem does not have a feasible tension, our algorithm not only presents a cycle, C, with $d^+(C) < 0$, but also gives some information so that the modeler can estimate the maximum cost of repairing the network in order to have a network with a feasible tension.

This paper consists of three sections in addition to the introduction. Section 2 presents a brief outline of the tension rectification algorithm. A scaling implementation of the tension rectification algorithm is shown in Section 3. Finally, Section 4 presents a faster implementation of the algorithm described in Section 3.

2. The tension rectification algorithm

In this section, a brief outline of the tension rectification algorithm [6,15] is presented. Given an arbitrary potential, π, and its tension θ, let:

$$A^+ = \{(i,j) \in A | \theta_{ij} < l_{ij}\},$$

$$A^- = \{(i,j) \in A | \theta_{ij} > u_{ij}\}.$$

If $A^+ = \phi = A^-$, tension θ is feasible. Otherwise any arc $(r,s) \in A^+ \cup A^-$ is selected. Then, Minty's Lemma [18] is applied using the following painting of the arcs $(i,j) \in A$: red (if $l_{ij} < \theta_{ij} < u_{ij}$), black (if $\theta_{ij} \leq l_{ij}$ and $\theta_{ij} < u_{ij}$), white (if $\theta_{ij} \geq u_{ij}$ and $\theta_{ij} > l_{ij}$), and green (if $l_{ij} = \theta_{ij} = u_{ij}$). It is obvious that (r,s) will be black or white.

If the outcome of Minty's lemma is a cycle, C, containing (r,s) terminate; this cycle has $d^+(C) < 0$. Otherwise, the outcome is a set, W, of nodes, such that $(r,s) \in (W,\overline{W}) = \{(i,j) \mid i \in W,\ j \notin W\}$ or $(r,s) \in (\overline{W},W) = \{(i,j) \mid i \notin W,\ j \in W\}$, where $\overline{W} = N - W$. In this case, the value, α, is computed as follows:

$$\alpha = \begin{cases} u_{ij} - \theta_{ij}, & \text{if } (i,j) \in (W,\overline{W}), \\ \theta_{ij} - l_{ij}, & \text{if } (i,j) \in (\overline{W},W), \\ \overline{\alpha}, & \text{otherwise,} \end{cases}$$

where:

$$\overline{\alpha} = \begin{cases} l_{rs} - \theta_{rs}, & \text{if } (r,s) \in A^+, \\ \theta_{rs} - u_{rs}, & \text{if } (r,s) \in A^-. \end{cases}$$

Then, the updated potentials are:

$$\pi'_i = \begin{cases} \pi_i, & \text{if } i \in W, \\ \pi_i + \alpha, & \text{if } i \in \overline{W}. \end{cases}$$

The tension rectification algorithm repeats with π' in which, after a finite number of iterations, arc (r,s) is finally removed from $A^+ \cup A^-$. These operations repeat for each arc in $A^+ \cup A^-$ and the algorithm can be run in $O(|A|(|A^+| + |A^-|)) \leq O(|A|^2) = O(m^2)$ time.

2.1. *Application of the feasible differential problem*

The minimum cost tension problem appears in many applications (Rockafellar [6, Chapter 7F]) concerning networks, such as timing of events, location of facilities, shared cost problems, and so on. Let c_{ij} denote the cost on arc (i,j). the minimum cost tension problem determines a feasible tension with minimum cost. *The minimum cost tension problem* is defined as follows:

$$\min \sum_{(i,j)\in A} c_{ij}\theta_{ij}$$

$$\text{s.t.} \quad \theta \text{ is a feasible tension.}$$

All minimum cost tension algorithms [2-5,7-9,19,20] start with a feasible tension. Thus, before solving the minimum cost tension problem, the feasible differential problem should be solved in order to find a feasible tension, or to diagnose that the problem does not have a feasible tension.

3. A scaling implementation of the tension rectification algorithm

In this section, a scaling implementation of the tension rectification algorithm is presented. Given a $\delta > 0$, we define tension θ as a *δ-feasible tension* if, for each $(i,j) \in A$:

$$l_{ij} - \delta < \theta_{ij} < u_{ij} + \delta. \quad (3)$$

Our algorithm starts out with δ large and drive δ toward zero. In each phase, the algorithm tries to find a δ-feasible tension using the input 2δ-feasible tension. The following lemma says that δ need not start out too big and end up too small.

Theorem 2. $\theta = 0$ is a $(U+1)$-feasible tension. Moreover, if l and u are integer and there exists a δ-feasible tension θ, such that $\delta \leq 1/m$, then, the feasible differential problem has a solution.

Proof. By $U = \max\{|l_{ij}|, |u_{ij}| \mid \text{for each}(i,j) \in A\}$, we get $l_{ij} - U \leq 0 \leq u_{ij} + U$ (for each arc $i \to j$), which means:

$$l_{ij} - (U+1) < 0 < u_{ij} + (U+1). \quad (4)$$

Let $\pi = 0$, which concludes $\theta = 0$ is a tension and, by Inequality (4), is a $(U+1)$-feasible tension.

Now, considering cycle C, by Eqs. (1) and (2), we get:

$$-d^+(C) = \sum_{(i,j)\in C^+} (\theta_{ij} - u_{ij}) + \sum_{(i,j)\in C^-} (l_{ij} - \theta_{ij}).$$

Tension θ is δ-feasible, so, by Inequality (3), we have:

$$-d^+(C) < \sum_{(i,j)\in C^+)} \delta + \sum_{(i,j)\in C^-} \delta \leq m\delta.$$

Thus, if we have a δ-feasible tension with $\delta \leq 1/m$, then, $d^+(C) > -m\delta \geq -1$. Since l and u are integer, $d^+(C)$ is integer too, which means $d^+(C) \geq 0$. Cycle C is arbitrary, so, Theorem 1 concludes that the feasible differential problem has a solution. □

Thus, our algorithm starts with $\delta = U$ and $x = 0$. In each phase, the input is a 2δ-feasible tension and the output is a δ-feasible tension. By Theorem 2, the algorithm runs in $O(\log(nU))$ phases. To explain a phase, we need the following definition.

Definition 1. Given a tension, θ, for each node, $i \in N$, value $\phi(i)$ is defined by the following:

$$\phi(i) = \max \begin{cases} l_{ij} - \theta_{ij}, & \text{for each outgoing arc } (i,j) \text{ of node } i, \\ \theta_{ri} - u_{ri}, & \text{for each incoming arc } (r,i) \text{ of node } i. \end{cases} \quad (5)$$

The next conclusion is a result of definitions, which presents a relationship among $\phi(i)$'s and δ-feasible tension.

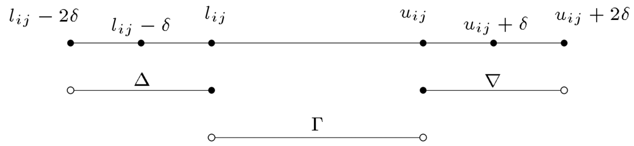

Figure 1. If $\theta_{ij} \leq l_{ij}$, then (i,j) is in the set Δ. If $l_{ij} < \theta_{ij} < u_{ij}$, then (i,j) is in the set Γ. Also, if $\theta_{ij} \geq u_{ij}$, then (i,j) is in the set ∇.

(a) (b)

Figure 2. If $\theta_{ji} > u_{ji}$ or $\theta_{ij} < u_{ij}$, then $\alpha(i) = \alpha(i) \cup \{j\}$.

Conclusion 1. Given a tension, θ. For each $i \in N$, $\phi(i) < \delta$ if and only if $l_{ij} - \delta < \theta_{ij} < u_{ij} + \delta$, for each $(i,j) \in A$.

Using Conclusion 1, we can work on $\phi(i)$'s in order to have a δ-feasible tension. For this purpose, we need the following definitions. Sets ∇, Δ and Γ are defined in the following way (Figure 1):

$$\Delta = \{(i,j) \in A \mid l_{ij} - 2\delta < x_{ij} \leq l_{ij}\},$$

$$\nabla = \{(i,j) \in A \mid u_{ij} \leq x_{ij} < u_{ij} + 2\delta\},$$

$$\Gamma = \{(i,j) \in A \mid l_{ij} < x_{ij} < u_{ij}\}.$$

Let $F(\delta) = \{i \in N \mid \delta \leq \phi(i) < 2\delta\}$, so, there is a relationship among sets $F(\delta), \Delta$ and ∇, which is as follows.

Conclusion 2. If $i \in F(\delta)$, then, at least one of the following occurs:

a) There is at least one outgoing arc (i,j) of node i with $\theta_{ij} < l_{ij}$.
b) There is at least one incoming arc (r,i) of node i with $\theta_{ri} > u_{ri}$.

Considering a node $i \in F(\delta)$, using Conclusion 2, we define node $\alpha(i)$ as follows.

Definition 2. For each outgoing arc (i,j) of node i with $\theta_{ij} < l_{ij}$ (resp. incoming arc (r,i) of node i with $\theta_{ri} > u_{ri}$), let $\alpha(i) = \alpha(i) \cup \{j\}$ (resp. $\alpha(i) = \alpha(i) \cup \{r\}$), see Figure 2(a) and (b).

In each phase of our algorithm, a 2δ-feasible tension, θ, should be changed to a δ-feasible tension, θ'. If $F(\delta) = \phi$, then θ is δ-feasible tension, and the current phase is finished. Otherwise, it selects a node, $i \in F(\delta)$, and labels the nodes using the labeling procedure (i) presented in Algorithm 1. The set of labeled nodes at

```
labeling procedure (i):
begin
  label i
  let LIST= {i};
  while LIST≠ φ do
   begin
    remove a node i from LIST;
    for each outgoing arc (i, j) of node i do
      if (i, j) ∈ ∇ and node j is unlabeled then label j, and add j to LIST;
    for each incoming arc (r, i) of node i do
      if (r, i) ∈ Δ and node r is unlabeled then label r, and add r to LIST;
   end;
  Let W be the set of labeled nodes;
end.
```

Algorithm 1. The labeling procedure.

```
Scaling tension rectification algorithm:
begin
  let π_i = 0 and α(i) = φ, for each i ∈ N
  let δ := U + 1;
  do until δ < 1/m;
    begin
      compute φ(i) for each i ∈ N;
      define F(δ) = {i ∈ N | δ ≤ φ(i) < 2δ};
      while F(δ) ≠ φ do
        begin
          remove a node i from F(δ);
          if there is one outgoing arc (i, j) of node i with θ_ij < l_ij then let α(i) = α(i) ∪ {j}
          else find one incoming arc (r, i) of node i with θ_ri > u_ri, and let α(i) = α(i) ∪ {r}
          labeling procedure(i);
          if α(i) ∩ W ≠ φ, then write "The problem does not have a solution" and break;
          else adjust π_i' s and θ_ij's by (8), (9) and update F(δ);
        end;
      let δ = δ/2;
      compute F(δ);
    end;
end.
```

Algorithm 2. The scaling tension rectification algorithm.

the end of the labeling procedure (i) is defined by W. The following theorem shows how it can be diagnosed when the feasible differential problem does not have a solution.

Theorem 3. Let W be the set of labeled nodes at the end of the labeling procedure (i). If $\alpha(i) \cap W \neq \phi$, then, the feasible differential problem does not have a solution.

Proof. By the labeling procedure (i), when $\alpha(i) \cap W \neq \phi$, there is a cycle, C, such that:

$$(i,j) \in \nabla, \quad \text{for each} \quad (i,j) \in C^+,$$

$$(i,j) \in \Delta, \quad \text{for each} \quad (i,j) \in C^-.$$

Thus, we get:

$$\sum_{(i,j)\in C^+} u_{ij} \leq \sum_{(i,j)\in C^+} \theta_{ij}, \tag{6}$$

and:

$$\sum_{(i,j)\in C^-} \theta_{ij} \leq \sum_{(i,j)\in C^-} l_{ij}. \tag{7}$$

The arc (i,j) or (j,i) with $j \in \alpha(i)$ is in cycle C, so, by Figure 2, one of Inequalities (5) or (6) is strict. Thus, by Eq. (1), we have:

$$\sum_{(i,j)\in C^+} u_{ij} - \sum_{(i,j)\in C^-} l_{ij} < 0,$$

which means, using Theorem 1, the feasible differential problem does not have a solution. □

Using Theorem 2 and Conclusion 2, we get the following conclusion, which shows how the feasibility of the feasible differential problem can be diagnosed.

Conclusion 3. If l and u are integer and $F(\delta) = \phi$, such that $\delta \leq 1/m$, then, the feasible differential problem has a solution.

Algorithm 2 presents our method, which selects a node, $i \in F(\delta)$, then removes node i from set $F(\delta)$ by adjusting π_i' s and θ_{ij}'s, as follows.

$$\pi'_i = \begin{cases} \pi_i - \delta, & \text{if } i \in W, \\ \pi_i, & \text{if } i \in \overline{W}. \end{cases} \tag{8}$$

By Eq. (7), θ_{ij}'s are changed according to:

$$\theta'_{ij} = \begin{cases} \theta_{ij} + \delta, & \text{if } (i,j) \in (W, \overline{W}), \\ \theta_{ij} - \delta, & \text{if } (i,j) \in (\overline{W}, W), \\ \theta_{ij}, & \text{otherwise}. \end{cases} \tag{9}$$

The next lemma proves that node i leaves set $F(\delta)$ using Eqs. (7) and (8).

Lemma 1. Suppose that $\alpha(i) \cap W \neq$ at the end of the labeling procedure (i). After adjusting π_i' s and θ_{ij}'s, according to Eqs. (7) and (8), we have $\phi(i) < \delta$.

Proof. Let $\overline{W} = N - W$. At the end of the labeling procedure (i), we have (Figure 3):

$(i,j) \in \nabla$, for each outgoing arc (i,j) of node i with $j \in W$,

$(r,i) \in \Delta$, for each incoming arc (r,i) of node i with $r \in W$,

$(i,q) \in \Delta$ or Γ, for each outgoing arc (i,q) of node i with $q \in \overline{W}$,

$(p,i) \in \nabla$ or Γ, for each incoming arc (p,i) of node i with $p \in \overline{W}$.

We consider the following cases in order to compute $\phi(i)$, with regard to θ', computed by Eq. (8):

(i) The outgoing arcs of node i.

(i-1) If (i,j) is an outgoing arc of node i with $j \in W$, then, by Figure 3, $(i,j) \in \nabla$, so, we get $u_{ij} \le \theta_{ij}$. By Eq. (8), $\theta'_{ij} = \theta_{ij}$, which means by $l_{ij} \le u_{ij}$, $l_{ij} - \theta'_{ij} \le 0 < \delta$.

(i-2) If (i,q) is an outgoing arc of node i with $q \in \overline{W}$, then, by Figure 3, $(i,q) \in \Delta$ or Γ, so, by Figure 1, we have $l_{iq} - 2\delta < \theta_{iq} < u_{iq}$. By Eq. (8), $\theta'_{iq} = \theta_{iq} + \delta$, so, by Figure 1, $l_{iq} - \delta < \theta'_{iq} < u_{iq} + \delta$, which means $l_{iq} - \theta'_{iq} < \delta$.

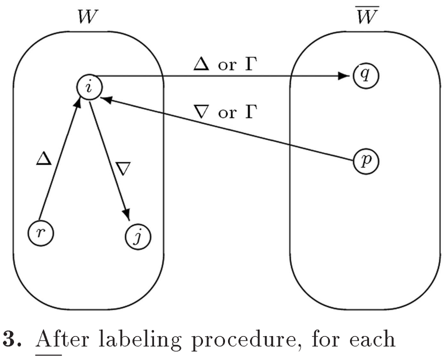

Figure 3. After labeling procedure, for each $(i,q) \in (W, \overline{W})$, we have $(i,q) \in \Delta \cup \Gamma$. Also, for each $(p,i) \in (\overline{W}, W)$, we have $(p,i) \in \nabla \cup \Gamma$.

(ii) The incoming arcs of node i.

(ii-1) If (r,i) is an incoming arc of node i with $r \in W$, then, by Figure 3, $(r,i) \in \Delta$, so, we get $\theta_{ri} \le l_{ri}$. By Eq. (8), $\theta'_{ri} = \theta_{ri}$, which means, by $l_{ri} \le u_{ri}$, $\theta'_{ri} - u_{ri} \le 0 < \delta$.

(ii-2) If (p,i) is an incoming arc of node i with $p \in \overline{W}$, then, by Figure 3, $(p,i) \in \nabla$ or Γ, so, by Figure 1, we have $l_{pi} < \theta_{pi} < u_{pi} + 2\delta$. By Eq. (8), $\theta'_{pi} = \theta_{pi} - \delta$, so, by Figure 1, $l_{pi} - \delta < \theta_{pi} < u_{pi} + \delta$, which means $\theta_{pi} - u_{pi} < \delta$.

By Definition 1 and cases (i-1),(i-2), (ii-1), and (ii-2), we get $\phi(i) < \delta$. □

Hence, by Lemma 1, node i leaves $F(\delta)$. In order to show that a phase finishes after finite iterations, we prove that the method does not enter a new node to set $F(\delta)$.

Lemma 2. After adjusting π_i' s and θ_{ij}'s, according to Eqs. (7) and (8), a new node does not add to set $F(\delta)$.

Proof. By Eq. (8), θ_{ij} is changed if $(i,j) \in (W, \overline{W})$ or $(i,j) \in (\overline{W}, W)$. Thus, we consider the following two cases:

Case-1. $j \in W$ with at least oneincoming arc of node j (Figure 4).

Case (1-1). If (i,j) is an incoming arc of node j with $i \in W$, then, by Figure 4, $(i,j) \in \Delta$ or Γ, so, we get $\theta_{ij} < u_{ij}$. By Eq. (8), $\theta'_{ij} = \theta_{ij} + \delta$, so $\theta'_{ij} < u_{ij} + \delta$, which means $\theta'_{ij} - u_{ij} < \delta$. Thus, by Definition 1, if node j is not in $F(\delta)$, then, by adjusting θ_{ij}, according to Eq. (8), node j will not be added to set $F(\delta)$.

Case (1-2). If (j,r) is an outgoing arc of node j with $r \in W$, then, by Figure 4, $(j,r) \in \nabla$ or Γ, so, we get $\theta_{jr} > l_{jr}$. By Eq. (8), $\theta'_{jr} = \theta_{jr} - \delta$, so, $\theta'_{jr} > l_{jr} - \delta$, which means $l_{jr} - \theta'_{jr} < \delta$. Thus, by Definition 1, if node j is not in $F(\delta)$, then, by adjusting θ_{jr} according to Eq. (8), node j will not be added to set $F(\delta)$.

Case-2. $i \in W$ with at least one outgoing arc of node i (Figure 5).

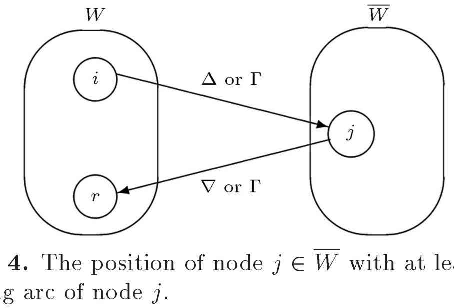

Figure 4. The position of node $j \in \overline{W}$ with at least one incoming arc of node j.

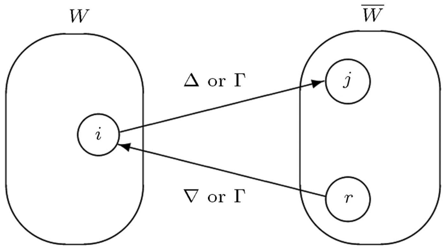

Figure 5. The position of node $i \in W$ with at least one outgoing arc of node i.

Using Figures 3 and 5 and cases (i-2) and (ii-2) in the proof of Lemma 1, if node i is not in $F(\delta)$, then, by adjusting θ_{ij}'s, according to Eq. (8), node i will not be added to set $F(\delta)$. $\square$

The next theorem computes the running time of the algorithm.

Theorem 4. The scaling tension rectification algorithm runs in $O(mn \log(nU))$ time.

Proof. Using Theorem 2, the number of phases is $O(\log(mU)) = O(\log(nU))$. Each phase, for a given $\delta > 0$, finishes, if $F(\delta) = \phi$. The algorithm selects a node, $i \in F(\delta)$, and changes it to $\phi(i) < \delta$ using Eqs. (7) and (8), which takes $O(m)$. By Lemma 2, during a phase, no new node will be added to $F(\delta)$, so, each phase needs $|F(\delta)| \leq n$ iterations. Thus, each phase runs in $O(mn)$. $\square$

Now, we show how the information of our algorithm is used to estimate maximum cost for repairing the network in order to have a feasible tension.

Theorem 5. For a given δ and $i \in F(\delta)$, if $\alpha(i) \cup W \neq \phi$, then, the network can be repaired by, at most, $2m\delta$ changes in bounds in order to have a feasible tension.

Proof. $\alpha(i) \cup W \neq \phi$ says the feasible differential problem does not have a feasible tension. Using the method of the algorithm, a 2δ-feasible tension is computed in the last phase. Thus, we have a tension, θ, with $l_{ij} - 2\delta < \theta_{ij} < u_{ij} + 2\delta$ (for each $(i,j) \in A$), which means that $m(2\delta)$ is an upper bound on the number of changes in bounds (increasing u_{ij}'s or decreasing l_{ij}'s by 2δ), so that θ be a feasible tension. $\square$

4. Reducing the number of phases to $O(\log\ U)$ and finding a feasible tension

At the end of Algorithm 2, we only can diagnose if the feasible differential problem has a solution, but do not find a feasible solution. In this section, we reduce the number of phases to $O(\log\ U)$ and compute a feasible solution (if it exists). Let $\overline{U} = 2^{\lfloor \log U \rfloor + 1}$, so, $\overline{U} > U + 1$, which means, by Theorem 2, $\theta = 0$ is a $\overline{U}$-feasible tension.

Lemma 3. Let l and u be integer. Supposing that the starting δ is $\delta = \overline{U}$, then:

a) If there is a 1-feasible tension, then the feasible differential problem has a solution.

b) Each 1-feasible tension is a feasible tension.

Proof. The starting values are $\delta = 2^{\lfloor \log U \rfloor + 1}$, $\theta_{ij} = 0$ (for each $(i,j) \in A$), and $\pi_i = 0$ (for each $i \in N$). Thus, initially, δ is integer and a multiplier of 2. In each iteration, δ is reduced to $\delta/2$, which means the values of all δ's are integer if $\delta \geq 1$. During each iteration, the values of π_i's and θ_{ij}'s are updated by Eqs. (7) and (8), so, they are integer if $\delta \geq 1$.

Supposing that we have a 1-feasible tension, θ, so, $l_{ij} - 1 < \theta_{ij} < u_{ij} + 1$, for each $(i,j) \in A$. Thus, for each $(i,j) \in A$, we get:

$$u_{ij} - \theta_{ij} > -1 \Rightarrow u_{ij} - \theta_{ij} \geq 0$$

$$(\text{because} \quad u_{ij} - \theta_{ij} \quad \text{is integer}),$$

$$\theta_{ij} - l_{ij} > -1 \Rightarrow \theta_{ij} - l_{ij} \geq 0$$

$$(\text{because} \quad \theta_{ij} - l_{ij} \quad \text{is integer}).$$

Therefore, $l_{ij} \leq \theta_{ij} \leq u_{ij}$, for each $(i,j) \in A$, which means each 1-feasible tension is a feasible tension. $\square$

The next theorem shows that the algorithm can be run in $O(mn \log U)$ time using Lemma 3.

Theorem 6. If the scaling tension rectification algorithm starts with $\delta = \overline{U}$, then, it runs in $O(mn \log U)$ time.

Proof. By Lemma 3, each 1-feasible tension is a feasible tension, so, the number of phases is $O(\log \overline{U}) = O(\log U)$. By the proof of Theorem 4, each phase runs in $O(mn)$. $\square$

Example 1. In this example, we present the implementation of our algorithm for the network in Figure 6. We have $U = 5$, so, the starting values are $\delta =$

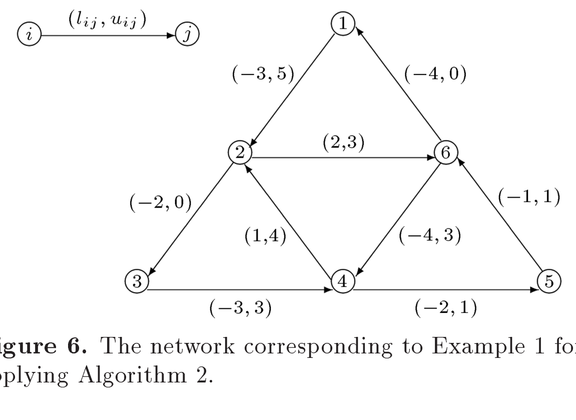

Figure 6. The network corresponding to Example 1 for applying Algorithm 2.

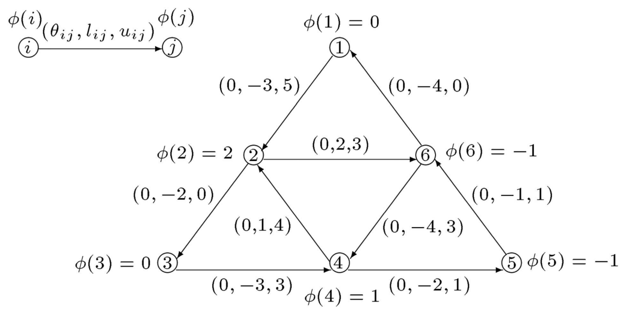

Figure 7. The initial values for $\phi(i)$'s and θ_{ij}'s in Example 1.

$2^{\lfloor \log 5 \rfloor} + 1 = 8$, and $\pi_i = 0$ for each $i \in N$ (i.e. $\theta_{ij} = 0$, for each $(i,j) \in A$). The starting $\phi(i)$'s are shown in Figure 7: $\phi(1) = 0$, $\phi(2) = 2$, $\phi(3) = 0$, $\phi(4) = 1$, and $\phi(5) = \phi(6) = -1$. Hence, we have $F(8) = \{i \in N \mid 8 \leq \phi(i) < 16\} = \phi$ and $F(4) = \{i \in N \mid 4 \leq \phi(i) < 8\} = \phi$, so, we let $\delta = 2$ and get $F(2) = \{i \in N \mid 2 \leq \phi(i) < 4\} = \{2\}$. For the outgoing arc $(2,6)$, we have $\theta_{26} = 0 < l_{26} = 2$, so $\alpha(2) = 6$ and a labeling procedure (2) should be done, so, we first label node 2. For the outgoing arc $(2,3)$, we have $\theta_{23} = 0 = u_{23}$, which means $(2,3) \in \nabla$, so, we label node 3. For the incoming arc $(4,2)$, we have $\theta_{42} = 0 < l_{42} = 1$, which means $(4,2) \in \Delta$, so, we label node 4. Other nodes can not be labeled. We have $W = \{2,3,4\}$, and $\overline{W} = \{1,5,6\}$, so, $\alpha(2)$ is not in W. Hence, by Eqs. (7) and (8), we get $\pi_2 = \pi_3 = \pi_4 = 0 - 2 = -2$, $\pi_i = 0$ (for other $i \in N$), $\theta_{26} = 0 + 2 = 2$, $\theta_{45} = 0 + 2 = 2$, $\theta_{12} = 0 - 2 = -2$, $\theta_{64} = 0 - 2 = -2$ and $\theta_{ij} = 0$ for other $(i,j) \in A$. The updated values of $\phi(i)$'s are $\phi(1) = \phi(2) = \phi(3) = 0$, $\phi(4) = 1$ and $\phi(5) = \phi(6) = -1$.

Now, we have $F(2) = \{i \in N \mid 2 \leq \phi(i) < 4\} = \phi$, so we let $\delta = 1$, and get $F(1) = \{i \in N \mid 1 \leq \phi(i) < 2\} = \{4,5\}$. By selecting node 4, updated values of π_i's are $\pi_2 = \pi_3 = -2$, $\pi_4 = -3$, $\pi_5 = -1$ and $\pi_i = 0$ (for other $i \in N$). Updated values θ_{ij}'s are $\theta_{42} = \theta_{56} = 1$, $\theta_{34} = -1$, $\theta_{64} = -3, \theta_{26} = 2, \theta_{45} = 2$, $\theta_{12} = -2$ and $\theta_{ij} = 0$, for other $(i,j) \in A$. The updated values of $\phi(i)$'s are $\phi(1) = \phi(2) = \phi(3) = \phi(4) = \phi(6) = 0$ and $\phi(5) = 1$.

Thus, $F(1) = \{i \in N \mid 1 \leq \phi(i) < 2\} = \{5\}$. For the incoming arc $(4,5)$ of node 5, we have $\theta_{45} = 2 > u_{45} = 1$, so, $4 \in \alpha(5)$ and the labeling procedure (5) should be done, which gives $W = \{5,6,2,4,3\}$, and $\alpha(5) \cap W \neq \phi$. Therefore, the problem does not have a solution. Note, in Cycle $C:\ 5-6-2-4-5$, we have $d^+(C) = (1+1) - (2+1) = -1 < 0$.

Now, the maximum cost is estimated for repairing this network in order to have a feasible tension. We have $F(2) = \emptyset$, but $F(1) \neq \emptyset$. Thus, by Theorem 5, an estimation of the maximum cost of repair is $2\delta m = 2(1)(9) = 18$. Of course, we have a lower estimation as follows:

$$\theta_{42} = 1 = l_{42}, \qquad \theta_{56} = 1 = u_{56},$$

$$l_{34} = -3 < \theta_{34} = -1 < u_{34} = 3,$$

$$l_{64} = -4 < \theta_{64} = -3 < u_{64} = 3, \qquad \theta_{26} = 2 = l_{26},$$

$$\theta_{45} = 2 > u_{45} = 1,$$

$$l_{12} = -3 < \theta_{12} = -2 < u_{12} = 5,$$

$$\theta_{61} = 0 = u_{61}, \qquad \text{and} \qquad \theta_{23} = 0 = u_{23}.$$

Thus we only need to change the upper bound arc $(4,5)$ by 1 unit (i.e. u_{45} should be increased to 2) in order to repair the network.

Acknowledgements

I would like to thank two anonymous referees for their valuable suggestions.

References

1. Berge, C. and Ghouila-Houri, A., *Programming, Games and Transportation Networks*, Wiley, New York (1962).
2. Ghouila-Houri, A. "Flows and tension in a graph" [Flots et tension dans un graph], Ph.D Thesis, Gauthier-Villars, Paris (1964).
3. Pla, J.M. "An out-of-kilter algorithm for solving minimum cost potential problems", *Mathematical Programming*, **1**, pp. 275-290 (1971)
4. Hadjiat, M. "Penelope's graph: a hard minimum cost tension instanc", *Theoretical Computer Science*, **194**, pp. 207-218 (1998).
5. Hamacher, H.W. "Min cost tension", *Journal of Information & Optimization Sciences*, **6**(3). pp. 285-304 (1985).
6. Rockafeller, R.T., *Network Flows and Monotropic Optimization*, John Wiley and Sons (1984).
7. Hadjiat, M. and Maurras, J.F. "A strongly polynomial algorithm for the minimum cost tension problem", *Discrete Mathematics*, 165/166, pp. 377-394 (1997).
8. Ghiyasvand, M. "An O(m(m+nlogn)log(nC))-time algorithm to solve the minimum cost tension problem", *Theoretical Computer Science*, **448**, pp. 47-55 (2012).
9. Ghiyasvand, M. "A polynomial-time implementation of Pla's method to solve the MCT problem", *Advances in Computational Mathematics and Its Applications*, **1**(2), pp. 104-109 (2012).
10. Ahuja, R.K., Hochbaum, D.S. and Orlin, J.B. "Solving the convex cost integer dual network flow problem", *Management Science*, **49**, pp. 950-964 (2003).
11. Bachelet, B. and Duhamel, C. "Aggregation approach for the minimum binary cost tension problem", *European Journal of Operations Research*, **197**, pp. 837-841 (2009).

12. Bachele, B. and Mahey, P. "Minimum convex-cost tension problems on series-parallel graphs", *RAIRO Operation Research*, **37**(4), pp. 221-234 (2003).

13. Bachele, B. and Mahey, P. "Minimum convex pircewise linear cost tension problem on quasi-k series-parallel graphs", *4OR: Quarterly Journal of European Operations Research Societies*, **2**(4), pp. 275-291 (2004).

14. Guler, C. "Inverse tension problems and monotropic optimization", *WIMA Report* (2008).

15. Goh, C.J. and Yang, X.Q., *Duality in Optimization and Variational Inequalities*, Taylor and Francis, London (2002).

16. Gabow, H.N. "Scaling algorithms for network problems", *Journal of Computer and System Science*, **31**, pp. 148-168 (1985).

17. Ahuja, R.K. Magnanti, T.L. and Orlin, J.B., *Network Flows: Theory, Algorithms, and Applications*, Prentice-Hall, Englewood Cliffs, NJ (1993).

18. Minty, G.J. "On the axiomatic foundations of the theories of directed linear graphs", *Electrical Networks and Programming, Journal of Mathematics and Mechanics*, **15**, pp. 485-520 (1966).

19. Hadjiat, M. and Maurras, J.F. "Duality between flow and tension", Actes des Troisiemes Journees du Groupe MODE, Berst, France (1995).

20. Maurras, J.F. "The maximum cost tension problem", *Proc. Conf.*, European chapter on combinatorial optimization (ECCO VII), Italy (1994).

Biography

Mehdi Ghiyasvand was born in Hamedan, Iran, in 1975. He obtained his MS and PhD degrees in Operations Research from Tehran University, Iran and spent a post-doctoral term at MIT University, USA. He is now Associate Professor at Bu-Ali Sina University, Hamedan, Iran.