Scientific
Research

Balanced Min Cost Flow on Skew Symmetric Networks with Convex Costs

Henning Soller

Institut für Theoretische Physik, Heidelberg, Germany
Email: H.Soller@ThPhys.Uni-Heidelberg.DE

ABSTRACT

We consider the solution of matching problems with a convex cost function via a network flow algorithm. We review the general mapping between matching problems and flow problems on skew symmetric networks and revisit several results on optimality of network flows. We use these results to derive a balanced capacity scaling algorithm for matching problems with a linear cost function. The latter is later generalized to a balanced capacity scaling algorithm also for a convex cost function. We prove the correctness and discuss the complexity of our solution.

Keywords: Matching; Skew Symmetric Network; Convex Cost Function; Optimization

1. Introduction

Skew symmetric networks have become an important tool for the efficient solution of matching problems [1]. Going over from a matching problem to a problem of flow optimization often allows for simplification and speed-up of solution algorithms. Typical matching problems do not only include maximizing a matching but often additionally minimizing the costs for the participants involved. Typical examples include minconvex-problems as studied in [2,3]. For such minconvex problems a convex function in the number of matchings has to be minimized for each participant. So far the solution of such problems using skew symmetric networks has not been demonstrated.

In this paper we therefore consider the problem of minimizing a separable convex objective function over a skew-symmetric network with a balanced flow. This problem can be mapped on the aforementioned matching problems and allows for an efficient solution of the latter. Specifically, we will derive a balanced capacity scaling algorithm incorporating the additional problem of a convex cost function.

In Section 2 we will shortly review skew symmetric networks and their connection to matchings. Section 3 will be devoted to a short summary of necessary results for optimality. In Section 4 we will present the balanced capacity scaling algorithm for a linear cost function going over in Section 5 to the description of the algorithm for a convex cost function. In Section 6 we will discuss some possible improvements of the aforementioned algorithm. We will conclude and sum up our results in Section 7.

2. Skew Symmetric Networks and Matchings

A *graph* $G = (V, E)$ is a pair of disjoint, finite sets E and V corresponding to the edges and vertices of the graph. If the edges between vertices (ij) are directed we have a *directed graph*. This allows for the definition of a network.

Definition: network

Let $G = (V, E)$ be a directed graph and $v, w : E \to \mathbb{R}$ two functions. We call the triplet (G, v, w) a network and the functions v and w the upper and lower capacity bound.

Definition: skew symmetric network [4]

A skew symmetric network is a network $N = (G, \mathrm{cap}, l)$ with the following properties

- the vertices of N contain a source s and a drain t and two sets of vertices $X = \{x_1, \cdots, x_n\}$ and X' such that a bijection $\pi : V \to V'$ exist.
- N contains edges $(sx_i), (x'_i t), \forall i, 1 \le i \le n$
- the other edges appear pairwise between X and X' meaning if $(x_i x'_j)$ exist then so does $(x'_i x_j)$ and vice versa
- the capacities obey $l(sx_i) = l(x'_i t), l(x_i x'_j) = l(x'_j x_i)$ and $\mathrm{cap}(sx_i) = \mathrm{cap}(x'_i t)$, $\mathrm{cap}(x_i x'_j) = \mathrm{cap}(x'_j x_i)$ for all edges in the network

Definition: flownetwork [5, p. 154]

Let N be a network and s and t are vertices in $V(N)$ such that s is connected to t via edges in $E(N)$. A flow is a mapping $x : E(N) \to \mathbb{R}$. If a flow is defined on N the network is called a flow network.

The flow on the network should later allow to map it to a matching. Therefore additional constraints are necessary. The excess is defined as

$$e(i) = \sum\nolimits_{(ji) \in E(N)} x_{ji} - \sum\nolimits_{(ij) \in E(N)} x_{ij} \qquad (1)$$

This allows to define an admissible flow.

Definition: admissible

A flow on a network is admissible if
$l(ij) \le x_{ij} \le \mathrm{cap}(ij)$ and $e(s) = -e(t)$.

Definition: balanced

A flow on a skew symmetric network is balanced if $x_{ij'} = x_{ji'}, x_{ij'} \in \mathbb{N}$ and for all loops $x_{ii'}$ is even.

An admissible flow can always be turned into a circulation, meaning a flow where no excess is found on any vertex. We just have to introduce a vertex from t to s which has a flow value of $e(t)$. Consequently is does not matter whether we consider circulation problems or flows.

Admissible and balanced flows correspond to matchings that we want to define below.

Definition: matching [6, p. 213]

Let $G = (V, E)$ be a graph, $a, b : V \to \mathbb{N}$, $c, d : E \to \mathbb{N}$. A graph $M = (V, E_M)$ with $E_M \subseteq E$ is called a matching if at least $a(i)$ edges and at most $b(i)$ edges in M are incident with the vertex i in V and for every $(ij) \in E$ i is incident with j at least $c(ij)$ times and at most $d(ij)$ times.

Theorem: correspondence

Every balanced admissible flow on a skew symmetric network corresponds to a matching on the corresponding graph with $a(i) = l(si), b(i) = \mathrm{cap}(si)$, $c(ij) = l(ij'), d(ij) = \mathrm{cap}(ij')$.

Proof: in [7], pp. 35-36

This correspondence is illustrated in **Figure 1** showing a graph with a matching and the corresponding skew symmetric flow network.

This first section summarized the previously known results on the correspondence between matchings and flow optimization which allows for efficient algorithms [5, p. 207].

3. Optimality of Network Flows

Since we have now reviewed the mapping between matchings and network flows we want to state several important results on the optimality of network flows which can be directly carried over. e.g. a maximal admissible balanced flow on the skew symmetric network corresponds to a maximal matching [7]. We start with the following definition.

Definition: restnetwork [7, p. 21]

Let N be a flownetwork and x the flow on it. Then the residual capacity corresponding to x is given by:

$$\mathrm{rescap}(ij) := \begin{cases} \mathrm{cap}(ij) - x_{ij}, & \text{for } (ij) \in E(N) \\ x_{ij}, & \text{for } \overline{(ij)} \in E(N) \\ 0, & \text{otherwise} \end{cases} \qquad (2)$$

$\overline{(ij)}$ is the backward edge, when the flow is negative. The edges (ij) with $\mathrm{rescap}(ij) > 0$ together with the vertices that coincide with an edge form the restnetwork $N(x)$.

So far we have only introduced the correspondence of network flows and matchings. However, we want to compute optimal matchings with respect to a cost function. This means that we have to consider problems on networks N of the type

$$\min C(x) = \sum_{(ij) \in E(N)} c_{ij} x_{ij}$$
$$l(ij) \le x_{ij} \le \mathrm{cap}(ij) \qquad (3)$$
$$x_{ij'} = x_{ji'}, x_{ij} \text{ integer}$$
$$e(i) = 0, \forall i \in V(N)$$

The additional complication compared to maximum balanced flow problems is the cost function. Therefore we want to introduce the necessary framework in order to deal with it. We start by the following
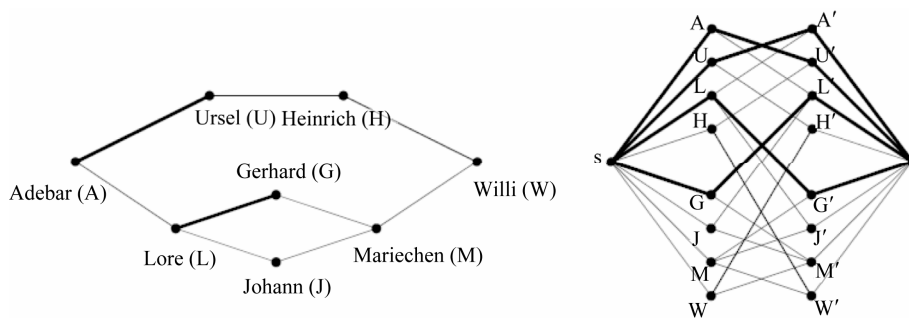

Figure 1. The left side shows a graph with vertices that are matched once (thick lines) and not matched (thin lines). On the left side the corresponding skew symmetric flow network is shown with thick lines corresponding to flows of 1.

Definition: potential, reduced costs [8]

The potential function $\pi : V \to \mathbb{R}$ associates with each vertex $i \in V$ a number $\pi(i)$, the potential i.

The reduced costs of an edge are defined as $c_{ij}^{\pi} = c_{ij} - \pi(j) + \pi(i)$.

The *length of a path* is then obviously defined as the sum of the reduced costs of the individual edges. The shortest valid path between two points in a network connects the two points via a valid path and the path has minimum length.

Corollary: [8]

For the reduced costs on a network N any path p from a vertex k to a vertex l fulfills

$$\sum_{(ji) \in p} c_{ij}^{\pi} = \sum_{(ji) \in p} c_{ij}^{\pi} - \pi(l) + \pi(k) \qquad (4)$$

and for any circle K

$$\sum_{(ji) \in K} c_{ij}^{\pi} = \sum_{(ji) \in K} c_{ij}^{\pi} \qquad (5)$$

Obviously we shouldn't try to find a solution by enhancing or decreasing the flow on arbitrary edges but we need a good measure for distance.

Definition: skew symmetric distances

Let $P_{i,j}$ be the set of admissible paths in the restnetwork $N(x)$ with start-node $i \in V(N(x))$ and end-node $j \in V(N(x))$. Then we define the skew symmetric distance from i to j as

$$d(i,j) := \min\left\{ \infty, \min_{p \in P_{i,j}} \left\{ \sum_{(ij) \in p} c_{ij}^{\pi} \right\} \right\} \qquad (6)$$

We call $d(i)$ the distance to s and the corresponding function is called d.

Definition: symmetric distances

Let d be the set of skew symmetric distances on a flow network N then the symmetric distances are given by

The corresponding set is denoted by sd.

$$sd(i) = \frac{1}{2}\left(d(i) + d(i') \right) \qquad (7)$$

At this point we state a theorem that has been extensively used for proving optimality

Theorem: Reduced Cost Optimality [8]

Let x be an admissible flow then x is optimal if a potential π exists such that for all edges (ij) in the rest-network

$$c_{ij}^{\pi} \geq 0. \qquad (8)$$

The potential π is often called the *dual solution*. It also has a practical importance [9]. Let us assume we have a logistics company with several warehouses. The transport costs per unit load between the different warehouses correspond to the costs on the edges. Then the potential for an optimal solution corresponds to the costs per unit load for storing them in a warehouse.

We need two further lemmas concerning the optimality of network flows

Lemma:

Let x be a balanced flow on a network N that fulfills the reduced cost optimality with respect to a potential π. Furthermore sd denotes the symmetric distances with respect to the reduced costs c_{ij}^{π} then

i) the flow x fulfills the reduced cost optimality also with respect to the potential $\pi' = \pi + sd$

ii) the reduced costs $c_{ij}^{\pi'}$ are zero on the shortest valid paths p and p'.

Proof:

We prove both statements one after the other:

i) Since x fulfills the reduced cost optimality $c_{ij}^{\pi} \geq 0, \forall (ij) \in N(x)$. Furthermore since sd is calculated from the shortest valid paths we know:

$$sd(j) \leq sd(i) + c_{ij}^{\pi} \forall (ij) \in N(x) \qquad (9)$$

We use the definition of the reduced costs

$$sd(j) \leq sd(i) + c_{ij} + \pi(i) - \pi(j)$$
$$0 \leq c_{ij} + (\pi(i) + d(i)) - (\pi(j) + d(j)) \qquad (10)$$
$$\Leftrightarrow c_{ij}^{\pi'} \geq 0$$

ii) Let p be an st path and p' its bijection. For every edge (ij) in p, we have $d(j) = d(i) + c_{ij}^{\pi}$. The same holds for the bijection. Therefore we have

$$c_{ij}^{\pi'} = c_{ij}^{\pi} - d(j) + d(i) = 0$$
$$c_{j'i'}^{\pi} = c_{j'i'}^{\pi} - d(i') + d(j') = 0 \qquad (11)$$

Lemma:

Assume a flow x fulfills the reduced cost optimality on a flow network N. If we change the flow both on the shortest valid path p and its bijection p' by

$$\text{balcap}(p) := \min_{(ij) \in p} \left\{ \begin{array}{l} \left\lfloor \dfrac{\text{rescap}(ij)}{2} \right\rfloor \text{ for } (ij)' \in p \\ \text{rescap}(ij) \quad \text{otherwise} \end{array} \right\} \qquad (12)$$

we find a new flow x' which also fulfills the reduced cost optimality with respect to the potential $\pi' = \pi + sd$.

Proof:

From the lemma above we know that the reduced costs are zero on p and p' with respect to π'. Therefore the reduced cost optimality cannot be violated if we enhance the flow by the maximum allowed capacity as defined above.

Obviously, an important ingredient is the solution of the shortest valid path problem. This has been discussed in detail in [10] and its complexity is $O(m \log n)$, where m is the number of edges and n is the number vertices in the network.

We now get to the combination of the results of Sec-

tion 2 and 3 being an algorithm to compute optimal flows on skew-symmetric networks.

4. Balanced Capacity Scaling

In this Section we describe the balanced capacity scaling algorithm for a linear cost function as in Equation (3) with the additional assumption that the costs are always positive. We will later generalize our result to a convex cost function. The idea of capacity scaling is to look at subgraphs in the restnetwork with some minimum capacity and to optimize the flow on these subgraphs successively.

We denote this minimum capacity by Δ and we call each phase of the algorithm where Δ does not change a Δ-*scaling phase*. We define two sets

$$S(\Delta) = \left\{ i \in V(N) \middle| e(i) \geq \Delta \right\}$$
$$T(\Delta) = \left\{ i \in V(N) \middle| e(i) \leq -\Delta \right\} \tag{13}$$

We can first calculate the maximum balanced flow through the network using one of the algorithms in [7] and use the result b as $e(s) = -e(t) = b$. We now begin the algorithm with flow 0 and potential 0 such that the reduced cost optimality is fulfilled but the flow does not fulfill Equation (3) as far as the excess is concerned.

For correctly prescribed Δ initially $S(\Delta) = \{s\}, T(\Delta) = \{t\}$. We define $U := 1 + e(s)$ and start with $\Delta = 2^{\lfloor \log_2 U \rfloor}$. This is the largest U such that not initially $S(\Delta) = T(\Delta) = \{ \ \}$. Furthermore we define

$$c_{\max} = \sum_{(ij) \in E(N)} c_{ij} + 1 \tag{14}$$

We additionally need the following
Definition: Δ-restnetwork

The Δ-restnetwork $N(\Delta)$ is defined as the network formed when only taking the edges in a flownetwork N into account with $\mathrm{rescap}(ij) \geq \Delta$. If the original flow network was skew symmetric, obviously also $N(\Delta)$ is skew symmetric.

We now denote the balanced capacity scaling algorithm and afterwards prove its correctness. We state the algorithm in a form close to typical programming language.

begin
$\quad x := 0, \pi := 0, e(s) = -e(t) = b,$
$\quad e(i) = 0, \forall i \in V(N) / \{s,t\}$
$\quad \Delta := 2^{\lfloor \log_2 (1+b) \rfloor}$
$\quad$ while $\Delta \geq 1$ do
$\quad\quad$ begin
$\quad\quad\quad$ for every edge (ij) in the Δ-restnetwork
$\quad\quad\quad$ do

$\quad\quad\quad\quad$ if $\mathrm{rescap}(ij) \geq \Delta, c_{ij}^{\pi} < 0$ then
$\quad\quad\quad\quad\quad$ increase flow on (ij) by $\mathrm{rescap}(ij)$
$\quad\quad\quad\quad\quad$ increase flow on $(j'i')$ by $\mathrm{rescap}(ij)$
$\quad\quad\quad\quad\quad$ recalculate the $e(i)$'s
$$S(\Delta) = \left\{ i \in V(N) \middle| e(i) \geq \Delta \right\}$$
$$T(\Delta) = \left\{ i \in V(N) \middle| e(i) \leq -\Delta \right\}$$
$\quad\quad\quad\quad$ while $S(\Delta) \neq 0, T(\Delta) \neq 0$ do
$\quad\quad\quad\quad\quad$ begin
$\quad\quad\quad\quad\quad\quad$ take a vertex $k \in S(\Delta), l \in T(\Delta)$
$\quad\quad\quad\quad\quad\quad$ calculate the shortest path from l to k in
$\quad\quad\quad\quad\quad\quad$ $N(\Delta)$ or connect the two with an edge
$\quad\quad\quad\quad\quad\quad$ $(lk) = p$ with costs $c_{\max} + \pi(i) - \pi(j)$
$\quad\quad\quad\quad\quad\quad$ $\pi = \pi + sd$
$\quad\quad\quad\quad\quad\quad$ enhance the flux by $\min\{\mathrm{balcap}(p), \Delta\}$
$\quad\quad\quad\quad\quad\quad$ units on p and p'
$\quad\quad\quad\quad\quad\quad$ update $x, S(\Delta), T(\Delta), N(\Delta)$
$\quad\quad\quad\quad\quad$ end;
$\quad\quad\quad\quad$ $\Delta = \Delta/2$
$\quad\quad$ end;
end;

Theorem:
The balanced capacity scaling algorithm calculates a maximal flow x on a skew symmetric network N with minimal costs.

Proof:
We prove the theorem by induction in the Δ-scaling phases.

In the beginning we have $c_{ij} = c_{ij}^{\pi} \geq 0$ so that the reduced cost optimality is fulfilled. The network is skew symmetric in the beginning.

Now we assume the solution to be optimal in the 2Δ-scaling phase and go over to the Δ-scaling phase. New edges added in the Δ-scaling phase may have negative reduced costs. For them $\Delta \leq \mathrm{rescap}(ij) \leq 2\Delta$ holds and we can enhance the flow on them by $\mathrm{rescap}(ij)$ since the costs are negative. In this case they are not part of the Δ-restnetwork. Since the same has to hold for the bijection the network will be skew symmetric again. Consequently the reduced cost optimality is fulfilled after the first part.

In the second part we enhance the flow on shortest valid paths so that the reduced cost optimality will be fulfilled. If an edge (lk) is introduced with costs $c_{\max} + \pi(i) - \pi(j)$ the costs are so high that the edge will not be part of the final solution.

We only need to show that the algorithm obtains an admissible flow for $\Delta < 1$. However, for $\Delta < 1$ no vertex can exists with an excess greater than one. However, due to the correspondence theorem the problem can only have integer flow variations so that no vertex with an excess smaller than one can exist.

Therefore we have found an optimal solution which has to be maximal since $e(s)$ was assumed to be the maximal flow through the network.

We therefore have an algorithm to solve the Balanced Min Cost Flow Problem on skew symmetric networks. In the next section we want to discuss the additional complication of a problem with a convex cost functions.

5. Convex Costs

We first define the generalization of the problem in Equation (3) with a convex cost function:

$$\min C(x) = \sum_{(ij) \in E(N)} C_{ij}(x_{ij})$$

$$l(ij) \le x_{ij} \le \operatorname{cap}(ij) \tag{15}$$

$$x_{ij'} = x_{ji'}, x_{ij} \text{ integer}$$

$$e(i) = 0, \forall i \in V(N)$$

The function $C_{ij} : D \to \mathbb{R}$ is a mapping from $D \subseteq \mathbb{R}^+$ into the real numbers and we know that for all $x, y \in D, \alpha, \beta \in \mathbb{R}, \alpha + \beta = 1$

$$c_{ij}(\alpha x + \beta y) \le \alpha c_{ij}(x) + \beta c_{ij}(y) \tag{17}$$

For a balanced capacity scaling algorithm for the problem in Equation (15) we follow [11, pp. 556-560].

The idea is to approximate the convex cost function by linear interpolation. This interpolation can be improved step by step until it is exact for all integer values as illustrated in **Figure 2**.

In the figure in each step we double the number of points in between which we assume the function to be constant. The frequency polygon therefore becomes a better approximation of the original function until we approximate the function at each integer value.

Since only integer flow values need to be considered the solution will be exact. In every Δ-scaling phase only changes of the flow values by $\{-\Delta, 0, \Delta\}$ need to be considered. Therefore we define the capacity and cost function for a Δ-scaling phase as:

$$\operatorname{cap}_{\Delta}(ij') = \begin{cases} \Delta, \text{ if } \operatorname{cap}(ij') - x_{ij'} \ge \Delta \\ 0, \text{ otherwise} \end{cases}$$

$$\operatorname{cap}_{\Delta}(j'i) = \begin{cases} \Delta, \text{ if } x_{ij'} \ge \Delta \\ 0, \text{ otherwise} \end{cases}$$

$$cc_{ij'} = \frac{C_{ij'}(x_{ij'} + \Delta) - C_{ij'}(x_{ij'})}{\Delta} \tag{18}$$

$$cc_{j'i} = \frac{C_{ij'}(x_{ij'} - \Delta) - C_{ij'}(x_{ij'})}{\Delta}$$

In the previous algorithm we additionally defined $c_{\max}$ which we have to do here as well. No edge in the network can have costs of:
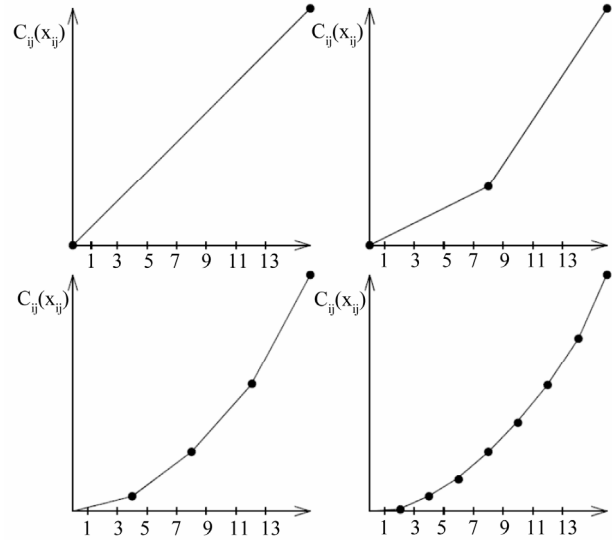

Figuare 2. successive approximation of a convex cost function.

$$cc_{\max} = \sum_{(ij) \in E(N)} C_{ij}(e(s)) + 1 \tag{19}$$

$e(s)$ is chosen again to be the maximum flow value through the network. We denote now the algorithm and prove its correctness for solving problems of the type as in Equation (15) afterwards.

begin
$\quad x := 0, \pi := 0, e(s) = -e(t) = b,$

$\quad e(i) = 0, \forall i \in V(N)/\{s, t\}$

$\quad \Delta := 2^{\lfloor \log_2(1+b) \rfloor}$

$\quad$ while $\Delta \ge 1$ do
$\quad \quad$ begin
$\quad \quad \quad$ for every edge (ij) in the Δ-restnetwork
$\quad \quad \quad$ do
$\quad \quad \quad$ if $c_{ij}^{\pi} < 0, c_{j'i'}^{\pi} \ge 0$ then
$\quad \quad \quad$ increase flow on (ij) and on $(j'i')$ by Δ
$\quad \quad \quad$ if $c_{ij}^{\pi} \ge 0, c_{j'i'}^{\pi} < 0$ then
$\quad \quad \quad$ reduce flow on (ij) and on $(j'i')$ by Δ
$\quad \quad \quad$ recalculate the $e(i)$'s
$\quad \quad$ $S(\Delta) = \{i \in V(N) | e(i) \ge \Delta\}$

$\quad \quad$ $T(\Delta) = \{i \in V(N) | e(i) \le -\Delta\}$

$\quad \quad$ while $S(\Delta) \ne 0, T(\Delta) \ne 0$ do
$\quad \quad \quad$ begin
$\quad \quad \quad$ take a vertex $k \in S(\Delta), l \in T(\Delta)$
$\quad \quad \quad$ calculate the shortest path from l to k in
$\quad \quad \quad$ $N(\Delta)$ or connect the two with an edge $(lk) = p$
$\quad \quad \quad$ with costs $cc_{\max} + \pi(i) - \pi(j)$
$\quad \quad \quad$ $\pi = \pi + sd$ enhance the flux by
$\quad \quad \quad$ $\min\{\operatorname{balcap}(p), \Delta\}$
$\quad \quad \quad$ units on p and p'

update $x, S(\Delta), T(\Delta), N(\Delta)$ and the reduced
costs
    end;
   $\Delta = \Delta/2$
  end;
end;

We have to prove that the algorithm is correct.

Theorem:

The balanced capacity scaling algorithm calculates a maximal flow x on a skew symmetric network with a convex cost function with minimal costs.

Proof:

We first show that the conversion of the problem into the algorithm is correct. In the beginning $S = \{s\}$. $T = \{t\}$ and $\Delta \leq |e(s)| = |e(t)| < 2\Delta$ so that not more than Δ flow units can be shifted in the first phase. Due to the choice of Δ this holds also for the next Δ-scaling phases $(\Delta \leq e(i) < 2\Delta \vee i \in N(\Delta))$. Consequently it is sufficient to only consider changes of the flow by $\pm \Delta$.

We now have to show that the reduced cost optimality is always fulfilled. In the beginning we have

$$cc_{ij} = \frac{C_{ij}(\Delta) - C_{ij'}(0)}{\Delta} \geq 0$$
$$cc_{j'i} = \frac{C_{ij'}(-\Delta) - C_{ij'}(0)}{\Delta} =: 0 \quad (20)$$

so that the reduced cost optimality is fulfilled.

Now let us assume that x is the flow after the 2Δ-scaling phase. We have the following cases for the flow in the Δ-scaling phase: i) $c_{ij}^\pi \geq 0, c_{ji}^\pi \geq 0$, ii) $c_{ij}^\pi < 0, c_{ji}^\pi \geq 0$, iii) $c_{ij}^\pi \geq 0, c_{ji}^\pi < 0$, iv) $c_{ij}^\pi < 0, c_{ji}^\pi < 0$.

The case iv) cannot occur for a convex cost function since we would have

$$\frac{C_{ij}(x_{ij} + \Delta) - C_{ij'}(x_{ij})}{\Delta} + \pi(i) - \pi(j) < 0$$
$$\frac{C_{ij}(x_{ij} - \Delta) - C_{ij}(x_{ij})}{\Delta} + \pi(i) - \pi(j) < 0 \quad (21)$$
$$\Rightarrow C_{ij}(x_{ij} + \Delta) - C_{ij}(x_{ij}) < C_{ij}(x_{ij}) - C_{ij}(x_{ij} - \Delta)$$

However, from the definition of a convex cost function we know

$$2C_{ij}(x_{ij}) \leq C_{ij}(x_{ij} - \Delta) + C_{ij}(x_{ij} + \Delta)$$
$$\Rightarrow C_{ij}(x_{ij} + \Delta) - C_{ij}(x_{ij}) \geq C_{ij}(x_{ij}) - C_{ij}(x_{ij} - \Delta) \quad (22)$$

and we have the disagreement.

The case i) fulfills the reduced cost optimality and we are left with ii) and iii). In case ii) we resolve this issue by enhancing the flow by Δ flow units. After the 2Δ-scaling phase we have

$$C_{ij}(x_{ij} + 2\Delta) - C_{ij}(x_{ij}) + 2\Delta\pi(i) - 2\Delta\pi(j) \geq 0 \quad (23)$$

and we have

$$C_{ij}(x_{ij} + \Delta) - C_{ij}(x_{ij}) + \Delta\pi(i) - \Delta\pi(j) < 0 \quad (24)$$

From the inequality from the 2Δ-scaling phase follows

$$C_{ij}(x_{ij} + 2\Delta) - C_{ij}(x_{ij} + \Delta) + \Delta\pi(i) - \Delta\pi(j)$$
$$+ C_{ij}(x_{ij} + \Delta) - C_{ij}(x_{ij}) + \Delta\pi(i) - \Delta\pi(j) \geq 0 \quad (25)$$

and the inequality follows since the last line of Equation (25) is smaller than zero by Equation (24) and consequently Equation (23) follows.

The reduced costs have to be identical on $(j'i')$ since the network is skew symmetric. We can treat case iii) completely analogous.

Everything we still need to show is that the reduced cost optimality is also fulfilled if we enhance the flow by balcap(p) and not by Δ. We consider $c_{ij}^\pi = 0$ and we obtain for the reduced costs

$$\Delta \cdot cc_{ij}^{neu} = C_{ij}(x_{ij} + \text{balcap}(p) + \Delta)$$
$$- C_{ij}(x_{ij} + \text{balcap}(p)) + \Delta\pi(i) - \Delta\pi(j) \quad (26)$$
$$\geq C_{ij}(x_{ij} + \Delta) - C_{ij}(x_{ij}) + \Delta\pi(i) - \Delta\pi(j)$$

Consequently the reduced cost optimality is still fulfilled. The same can be shown for the edge $(j'i')$.

At the end we will therefore obtain an admissible flow which is the solution to the Min Cost Flow Problem on a skew symmetric network with a convex cost function.

We will finally get to the complexity of this problem using the above algorithm.

Theorem

The balanced capacity scaling algorithm has a complexity of $O(\log_2 U \cdot m \cdot S(n,m))$ with the complexity $S(n,m)$ of computing a shortest valid path.

Proof:

At the end of each 2Δ-scaling phase we have $T(2\Delta) = \varnothing$ or $S(2\Delta) = \varnothing$ so that at most there is $2n\Delta$ of flow that can be shifted along in the next phase. Additionally we change the flow at the beginning of the Δ-scaling phase by at most $2m\Delta$ so that the total excess can be at most $2(n+m)\Delta$. In every iteration at least $\Delta/2$ flow units are shifted so that we will have at most $O(m)$ iterations of the shortest valid path algorithm so that the complexity is $O(mS(n,m))$. We can have at most $\log_2 U$ phases so that the complexity follows.

6. Possible Improvements

One can still improve the above algorithm by noticing that for $\Delta \geq 2$ there can be no edge (ij) with rescap$(ij) = 1$. In this case shortest paths in the network are always valid and we can use the simpler Dijkstra al-

gorithm for shortest paths. An implementation with Fibonacci heaps has complexity $O(n \log n + m)$ (see [11]) so that one only has to use the more involved shortest valid path algorithm in the last phase.

The author has also tried to implement other methods used in networks with a convex cost function like the Out-of-Kilter algorithm, the Relaxation Algorithm, Cancel-and-Tighten and the primal-dual algorithm. Inspite of the noticeable simplicity of the above algorithm none of the above provided real improvements in terms of complexity for the solution so that only the Balanced Capacity Scaling algorithm is presented here.

7. Conclusion

We have presented the first solution and complexity analysis of the Min Cost Flow Problem on a skew symmetric network with a convex cost function. The solution of this problem is of special relevance for the solution of matching problems with a convex cost function. The solution is based on a balanced capacity scaling algorithm.

8. Acknowledgements

REFERENCES

[1] W. Kocay and D. Stone, "Balanced Network Flows," *Bulletin of the Institute of Combinatorics and Its Applications*, Vol. 7, No. 1, 1993, pp. 17-32.

[2] N. Appolonio and A. Sebö, "Minconvex Factors of Prescribed Size in Graphs," Technical Report 145, Les Cahiers du Laboratoire Leibniz, 2006.

[3] A. Berger and W. Hochstättler, "Minconvex Graph Factors of Prescribed Size and a Simpler ´Reduction to Weighted f-Factors," *Electronic Notes in Discrete Mathematics*, Vol. 28, 2007, pp. 69-76. doi:10.1016/j.endm.2007.01.011

[4] C. Fremuth-Paeger and D. Jungnickel. "Balanced Network Flows. I. A Unifying Framework for Design and Analysis of Matching Problems," *Networks*, Vol. 33, No. 1, 1999, pp. 1-28. doi:10.1002/(SICI)1097-0037(199901)33:1<1::AID-NET1>3.0.CO;2-M

[5] B. Korte and J. Vygen, "Combinatorial Optimization," Springer Verlag, New York, 2000. doi:10.1007/978-3-662-21708-5

[6] D. Jungnickel, "Graphs, Networks and Algorithms," Springer Verlag, New York, 2004.

[7] C. Fremuth-Paeger, "Degree Constrained Subgraph Problems and Network Flow Optimization," Wißner Verlag, Augsburg, 1997.

[8] C. Weber, "Minimum Cost Flow Grundlagen und Erste Algorithmen," 2005. http://www.informatik.uni-trier.de/naeher/professur/courses/ws2005/exercises/seminar/ausarbeitungweber.pdf

[9] D. Bertsekas, "Network Optimization: Continuous and Discrete Models," Athena Scientific, Belmont, 1998.

[10] A. V. Goldberg and A. V. Karzanov, "Path Problems in Skew-Symmetric Graphs," *SODA*: *ACM Symposium on Discrete Algorithms*, Arlington, 23-25 January 1994, pp. 526-535.

[11] R. K. Ahuja, T. L. Magnanti and J. B. Orlin, "Network Flows: Theory, Algorithms and Applications," Prentice Hall, 1993.