枝重み付き一般グラフの最大マッチングの下限と線形時間近似アルゴリズム

梶谷 洋司[†] [*]　藤吉 邦洋[*]　Jun Dong Cho[‡]　Majid Sarrafzadeh[§]

[†]東京工業大学 電気電子工学科
[*]北陸先端科学技術大学院大学 情報科学研究科
[‡]Sung Kyun Kwan Univ., Korea
[§]Northwestern Univ., USA

枝重み付き一般構造グラフの最大マッチングの下限は点数 n が偶数のとき $\frac{w(E)}{(n-1)}$、奇数のとき $\frac{w(E)-w(I_v)}{n-2}$ であることを示す。但し、$w(E)$ は枝重み総和であり、$w(I_v)$ は最小の接続枝重み和である。証明は最大マッチングが性質4CYCLEを満たすことを用いる解析的な方法と線形アルゴリズムで構成する方法の二通りで示す。ここで4CYCLEとは、長さ4の交互サイクル上の変換でマッチング重み和が増大しないという性質である。後者で構成されるマッチングは必ずしも4CYCLEを満たさない。

和文キーワード：一般マッチング, 最大マッチング, 近似

A New Lower Bound and Linear Time Approximation Algorithm for Maximum Matching of General Edge-Weighted Graph

Yoji Kajitani[†] [*], Kunihiro Fujiyoshi[†], Jun Dong Cho[‡], and Majid Sarrafzadeh[§],

[†]Department of Electrical and Electronic Engineering,
Tokyo Institute of Technology

[*]School of Information Science,
Japan Advanced Institute of Science and Technology

[‡]Sung Kyun Kwan Univ., Korea

[§]Northwestern Univ., USA

For a general edge-weighted graph with n vertices, it is shown that the lower bound of the maximum matching is $\frac{w(E)}{(n-1)}$ if n is even and $\frac{w(E)-w(I_v)}{n-2}$ if n is odd where $w(E)$ and $w(I_v)$ denote the total edge weight sum and minimum of the total weights of the incidence edges. It is proved in two ways; one is to use the property 4CYCLE and the other by a linear time construction. 4CYCLE is the property that any transformation along the cycle of length 4 does not increase the weight of the matching. The matching constructed by the latter way does not necessarily satisfy 4CYCLE.

Key words : general matching, maximum matching, approximation

1 Introduction

Let $G = (V, E)$ be an edge-weighted graph without loops and multiple edges. Let $|V| = n$. For subset $E_s \subset E$, $w(E_s)$ denotes the sum of weights of edges of E_s, and $V(E_s)$ the set of end vertices of the edges of E_s. For subset $V_s \subset V$, $w(V_s)$ denotes the sum of weights of edges whose both end vertices are in V_s (edges of the induced subgraph defined by V_s). For two disjoint vertex subsets $V_1, V_2 \subset V$, $w(V_1; V_2)$ is the sum of weights of all edges each connecting a vertex of V_1 and that of V_2. Let I_v denotes the set of edges incident to a vertex v. A *matching* M is a subset of E such that no two edges of M are incident to a common vertex. A maximum matching is a matching the sum of whose edge weights is maximum of all matchings. Since we are concerned with maximum matchings, we assume without loss of generality that the weight of any edge is non-negative.

Even the simplest case, the cardinality matching in a bipartite graph, requires $O(n^{5/2})$ time for its exact solution [5]. In applications, sometimes we need only approximated solutions or simply the lower bound. However, these algorithms for exact solutions are too slow for the purpose. It is strongly hoped to develop practically useful algorithms, actually linear time algorithms, in trade-off to accuracy but with guaranteed approximation.

So far the best known lower bound for maximum matchings is $|E|/n$ for unweighted case[??]. Its proof is purely combinatorial and seems no generalization is possible for the weighted case.

As for the lower bound, we prove a result for general weighted case. It includes as a special case a strict improvement to the existing result for the unweighted case. The proof is made in two ways analytically and constructively. The property called 4CYCLE plays a key role which is a claim that an alternating cycle of length 4 does not increase the weight by the change. As for the fast algorithm, a linear time algorithm which provides a solution within the above mentioned lower bound is presented.

2 Theorems on Lower Bounds and Algorithms

2.1 Better Lower Bounds

Theorem 1 : Let M_{max} be a maximum matching of $G(V, E)$, then

$$\begin{aligned} \text{if } |V| \text{ is even} &\ : \ w(M_{max}) \geq \frac{w(E)}{|V|-1} \\ \text{if } |V| \text{ is odd} &\ : \ w(M_{max}) \geq \frac{w(E) - w(I_v)}{|V|-2} \qquad \forall v \in V. \end{aligned}$$

□

In the case of n:odd, the vertex v should be taken the one whose $w(I_v)$ is minimum to provide the lower bound tightest.

This theorem is exclusively described by the number of vertices and total weight of edges. However, if the graph is special or some additional information is available, we may have a tighter lower bound. One example is the following claim.

Corollary 1 : Let $G(V, E)$ be a bipartite with partition $V = V_L \cup V_R$. Then

$$\begin{aligned} \text{if } n \text{ is even} &\ : \ w(M_{max}) \geq \frac{w(E)}{\max(|V_L|, |V_R|)} \\ \text{if } n \text{ is odd} &\ : \ w(M_{max}) \geq \frac{w(E) - w(I_v)}{\max(|V_L|, |V_R|)} \qquad \forall v \in V. \end{aligned}$$

□

2.2 Linear Time Algorithms

Theorem 2 : There exists an $O(|V|^2)$ time algorithm to obtain a matching satisfying the claim in Theorem 1. □

Theorem 3 : If the graph is unweighted, there exists an $O(|V| + |E|)$ time algorithm to obtain a matching M satisfying

$$\begin{aligned} \text{if } |V| \text{ is even} &: \quad |M| \geq \frac{|E|}{|V|-1} \\ \text{if } |V| \text{ is odd} &: \quad |M| \geq \frac{|E| - |I_v|}{|V|-2} \qquad \forall v \in V. \end{aligned}$$

□

3 Analytical Proof

For simplicity of our discussion, we assume that $G(V, E)$ is a complete graph by adding zero-weighted edges if necessary. Furthermore, a maximum matching is assumed to consist of $n/2$ edges. A pair of two edges e_m and e'_m of a matching M belongs to two cycles (alternating cycles) each of length 4. Let one of them be e_m, e'_m, e_n, e'_n such that $e_n, e'_n \notin M$. If it holds

4CYCLE w.r.t. a matching edge pair (e_m, e'_m)

$$w(e_m) + w(e'_m) \geq w(e_n) + w(e'_n),$$

the transformed matching $(M - \{e_m, e'_m\}) \cup \{e_n, e'_n\}$ is not heavier than M. If M is a maximum matching, 4CYCLE is satisfied with respect to any pair of matching edges.

3.1 Proof of Theorem 1

The theorem is proved for the case of n being even. The case of n being odd is proved as its corollary.

Case : n is even Notice that $G(V, E)$ is complete and that M_{max} consists of $n/2$ edges. For each pair of edges in M_{max}, there are two alternating cycles. Since 4CYCLE is satisfied for every pair of matching edges, we have two inequalities for each pair of edges of M_{max}. If we list all of these $2 \times {}_{\frac{n}{2}}C_2$ inequalities, every edge of M_{max} belongs to $2(n/2 - 1)$ inequalities while every edge not in M_{max} appears exactly once. Therefore,

$$2(n/2 - 1) w(M_{max}) \geq w(E - M_{max}).$$

$$(n-2) w(M_{max}) \geq w(E) - w(M_{max}).$$

Thus

$$w(M_{max}) \geq \frac{w(E)}{n-1}.$$

Case : n is odd For $v \in V$, let G' be the graph obtained from G by deleting v. The edge set is $E' = E - I_v$. Let M'_{max} be a maximum matching in G'. Since the number of vertices of G' is even,

$$w(M_{max}) \geq w(M'_{max}) \geq \frac{w(E')}{n-2} = \frac{w(E) - w(I_v)}{n-2}.$$

□

3.2 Proof of Corollary 1

The corollary is not a special case of Theorem 1 but its proof follows a similar logic.

Assume that $n_L = |V_L| \leq n_R = |V_R|$.

Case : n is even Since the graph is assumed to be complete, there exists a maximum matching M_{max} such that it consists of n_L edges each connecting a vertex of V_L and that of V_R and $(n_R - n_L)/2$ edges connecting the vertices of V_R, the latter consists only of zero-weighted edges. The former edge set is denoted by M_{LR} and the latter by M_{RR}.

Since $V = V(M_{LR}) \cup V(M_{RR})$ is a partition,

$$w(V(M_{LR})) + w(V(M_{RR})) + w(V(M_{LR}); V(M_{RR})) = w(E) \tag{1}$$

Of two alternating cycles defined by each pair of matching edges in M_{LR}, one is a cycle exclusively consisted of the edges connecting the vertices in V_L and those in V_R. (The other cycle is one containing two zero-weighted edges which produce a trivial inequality by 4CYCLE.) Summing up all inequalities obtained by applying 4CYCLE to such cycles, we have

$$(n_L - 1)w(M_{LR}) \geq w(V(M_{LR})) - w(M_{LR}) \tag{2}$$

Moreover, applying 4CYCLE to all alternating cycles defined by one edge in M_{LR} and one in M_{RR}, we have $2 \times \left(n_L \times \frac{n_R - n_L}{2}\right)$ inequalities. Their sum leads to

$$2\left(\frac{n_R - n_L}{2} w(M_{LR}) + n_L w(M_{RR})\right) \geq w(V(M_{LR}); V(M_{RR})) \tag{3}$$

Taking the sum of (1),(2), and (3) with $w(M_{RR}) = w(V(M_{RR})) = 0$ and $w(M_{LR}) = w(M_{max})$, we conclude

$$n_R w(M_{max}) \geq w(E)$$

Case : n is odd The same technique of deleting one vertex as used in the proof of Theorem 1 leads the claim. □

4 Constructive Proof by $O(|E|)$ Time Algorithms for General Case

The algorithm constructs a matching, starting with the empty set by augmenting the cardinality one by one. Let the current matching be $M = \{e_1, e_2, \cdots, e_m\}$, $e_k = (t_k, b_k)$. For this current matching M, the temporary condition is defined as:

CurCondition(M): For a matching M,

$$w(M) \geq \frac{w(V(M))}{|V(M)| - 1}.$$

A vertex not in $V(M)$ is called the ***outside*** vertex and an edge whose end vertices are both outside is called the ***outside*** edge. Since graph $G(V, E)$ is assumed to be complete, any pair of vertices is considered to represent an edge.

Let (α, β) be any outside edge, Define the **swap-gain** $\delta(e_k)$ and $\delta'(e_k)$ of $e_k \in M$ to (α, β):

$$\begin{aligned}
\delta(e_k) &= w(t_k, \alpha) + w(b_k, \beta) - w(e_k) - w(\alpha, \beta) \\
\delta'(e_k) &= w(t_k, \beta) + w(b_k, \alpha) - w(e_k) - w(\alpha, \beta)
\end{aligned}$$

Though the notation is symmetric, we call the former and the latter *parallel gain* and *cross gain* respectively.

Find a maximum in $\{\delta(e_1), \delta'(e_1), \cdots, \delta(e_m), \delta'(e_m)\}$. Without loss of generality, we assume that it is a parallel gain $\delta(e_r) = w(t_r, \alpha) + w(b_r, \beta) - w(e_r) - w(\alpha, \beta)$.

If $\delta(e_r) \leq 0$, (α, β) is called **SwapNonPositive** We describe only the case when n is even since the case when n is odd is obtained as a corollary.

ALGORITHM for a weighted graph with $|V|$: even

(Initialization): $M = \emptyset$.

(Loop): Continue until $V(M) = V$ for arbitrary outside edge (α, β);

If condition **SwapNonPositive** is satisfied, let

$$M \Leftarrow M \cup \{(\alpha, \beta)\}$$

Otherwise, i.e. $\delta(e_r) > 0$, let

$$M \Leftarrow M \cup \{(t_r, \alpha), (b_r, \beta)\} - \{e_r\}$$

Lemma 1 : If CurCondition(M) is satisfied and (α, β) satisfies **SwapNonPositive**, the augmented matching $M' = M \cup \{(\alpha, \beta\}$ satisfies CurCondition(M').
Proof : From the assumption of CurCondition(M), we have

$$w(V(M)) \leq (2m - 1)w(M). \tag{4}$$

Since every swap-gain to (α, β) is non-positive,

$$\sum_{k=1}^{m} \delta(e_k) + \delta'(e_k) \leq 0.$$

Then, the above is equivalent to

$$w(V(M); \{\alpha, \beta\}) - 2w(M) - 2mw(\alpha, \beta) \leq 0. \tag{5}$$

The sum of (4) and (5) is

$$w(V(M)) + w(V(M); \{\alpha, \beta\}) \leq (2m + 1)w(M) + 2mw(\alpha, \beta).$$

or

$$w(V(M) \cup \{\alpha, \beta\}) \leq (2m + 1)w(M \cup (\alpha, \beta)).$$

This is CurCondition(M'). □

Lemma 2 : If the current matching M satisfies CurCondition(M) but the outside edge (α, β) does not satisfy **SwapNonPositive,** the augmented matching

$$M' = M \cup \{(t_r, \alpha), (b_r, \beta)\} - \{e_r\}$$

satisfies CurCondition(M').
Proof : Since $\delta(e_r)$ is maximum,

$$\sum_{k=1}^{m} \delta(e_k) + \delta'(e_k) \leq 2m\delta(e_r)$$

or

$$w(V(M); \{\alpha, \beta\}) - 2w(M) - 2mw(\alpha, \beta) \leq 2m\delta(e_r). \qquad (6)$$

The sum of (4) and (6) leads to

$$w(V(M)) + w(V(M); \{\alpha, \beta\}) \leq (2m + 1)w(M) + 2mw(\alpha, \beta) + 2m\delta(e_r)$$

or

$$w(V(M)) + w(V(M); \{\alpha, \beta\}) + w(\alpha, \beta) \quad \leq \quad (2m + 1)(w(M) + w(\alpha, \beta) + \delta(e_r)) - \delta(e_r)$$

By assumption of **SwapNonPositive** not being to hold,

$$\delta(e_r) = w(t_r, \alpha) + w(b_r, \beta)) - w(e_r) - w(\alpha, \beta) > 0.$$

This is CurCondition(M').

$$w(V(M) \cup \{\alpha, \beta\}) < (2m + 1)w(M \cup \{(t_r, \alpha), (b_r, \beta)\} - \{e_r\})$$

□

The initial matching trivially satisfies CurCondition($\emptyset$). Then by induction, CurCondition(M) is met when $|M| = n/2$, which is the theorem itself. **(End of proof for Theorem 1)**

Example: A constructing example will be shown.
$G(V, E)$ is given where

$V : 1, 2, \cdots, 8$, and

Edges and weights: $w(1,2) = 3, w(1,3) = 4, w(2,4) = 1, w(3,5) = 4, w(3,4) = 3, w(4,6) = 1, w(5,6) = 3, w(5,7) = 20, w(6,8) = 30, w(7,8) = 40, w(4,7) = 40$. Other edges are all weight 0.

Suppose it has been $M = \{(1,2), (3,4), (5,6)\}$ in order all by simple addition and now let $(\alpha, \beta) = (7,8)$. Since the swap gain of $(5,6)$ is positive, we have $M = \{(1,2), (3,4), (5,7), (6,8)\}$ with $W(M) = 56$ and ends. M satisfies the CurCondition(?) definitely as $56 \geq \frac{w(E)}{n-1} = \frac{149}{7} \sim 21.3$.

We observe that 4CYCLE is not satisfied in a length 4 cycle consisting of $(3,4), (5,7) \in M$ and $(3,5), (4,7) \notin M$. If we were to apply augmentation, we could have $M' = \{(1,2), (3,5), (4,7), (6,8)\}$ with $w(M') = 77$.

4.1 Proof of Theorem 2

The algorithm constructed in the proof of Theorem 1 runs in $O(E)$ time. To see if **SwapNonPositive** holds for each pair (α, β) is to find a maximum of $2m$ swap-gains and it takes $O(m)$. Either the maximum gain being positive or not, each time $|V(M)|$ increases by two. Hence the total complexity is $O(nm) = O(n^2) = O(|E|)$.

5 Constructive Proof by an $O(|V| + |E|)$ Time Algorithm for Unweighted Case

The lack of edges is an essential information to be used for reducing the time complexity. To represent this information, we introduce a property for a matching edge $e_m = (t, b)$ in M:

3LOCM(e_m) For any two distinct outside vertices α and β, there is no path between α and β of length three containing e_m.

In the following, an algorithm which constructs a matching starting with the empty set by augmenting one edge after another is sketched. Let M denote the current matching on the way.

ALGORITHM for unweighted graph with $|V|$: even

Step (simple addition): For each edge $e \in E$, if e is an outside edge, add it to M.

Step (augmentations): For each matching edge $e \in M$, if 3LOCM(e) is not satisfied, say there exist outside vertices α and β such that $(t, \alpha), (b, \beta) \in E$, augment M to

$$M \Leftarrow M \cup \{(t, \alpha), (b, \beta)\} - \{(t, b)\}.$$

End of the algorithm

Lemma 3 : The resultant matching M satisfies

$$|M| \geq \frac{|E|}{|V| - 1}.$$

Proof : First we prove that any matching edge $e \in M$ satisfies 3LOCM(e). Note that no outside vertex is created by any steps. Let M' be the new matching created from M by an augmentation inside **Step (augmentations)**. There are two new members e's. Each satisfies 3LOCM(e) in M' because there have been no outside edges with respect to M. Once a matching edge e is verified to satisfy 3LOCM(e), this property is kept in succeeding transformations in **Step (augmentations)**. Thus in the resultant matching M, every edge e in M satisfies 3LOCM(e).

Let $m = |M|$.

1. There is no edge between the outside vertices.
2. Make all $n - 2m$ outside vertices into $\frac{n-2m}{2}$ pairs arbitrarily. The number of edges between one of such pairs and one of matching edges is at most 2, since otherwise, those outside vertices had been augmented. Therefore, the total number of edges between $V(M)$ and the outside vertices is at most $2\frac{m(n-2m)}{2} = m(n - 2m)$.
3. The possible maximal subgraph induced by $V(M)$ is a complete graph. Therefore, the number of the edges in it is at most $\frac{2m(2m-1)}{2} = m(2m - 1)$.
4. Thus, $|E| \leq m(n - 2m) + m(2m - 1) = m(n - 1)$.
5. It suffices to show that $m \geq \frac{|E|}{n-1}$. This holds with equality.

□

Lemma 4 : The computational complexity of the algorithm is $O(|V| + |E|)$.

Proof : Following discussion is valid if the graph data is stored in adjacency list.

After the Step (simple addition), the total computation so far spent is $O(|V| + |E|)$.

In **Step (augmentations)**, the dominant term is the computation to test 3LOCM(e). For each vertex, make a list (*outside list*) of its adjacent outside vertices. Let a matching edge $e = (t, b)$. Check if the first elements in lists of t and b are distinct (distinction test). If yes, the 3LOCM(e) is not satisfied. If no, and if one list contains another element, 3LOCM(e) is not satisfied.

Investigating other cases, we conclude that 3LOCM(e) is satisfied if and only if either list is empty or both lists consist of identical one element. Anyway, at most one time distinction test and two times of next element search are enough for checking 3LOCM(e) for each matching edge e. Therefore, the complexity here is $O(|M|)$ for all matching edges.

Constructing the outside lists is possible in $O(|V| + |E|)$. Renewal of the outside lists is simply deletion of the outside vertices. Therefore if each current outside vertex maintains pointers which direct the address in every outside list, deletion is possible in constant time. Since each element in the list is deleted at most once, the total computation time is $O(|E|)$.

□

6 Conclusion

A local property of matching called 4CYCLE is abstracted and proved that a matching constructed so as to satisfy this condition sequentially is guaranteed in its approximation. Since 4CYCLE is a local condition, an algorithm that runs in liner time is possible. It is possible to think a property $2k$CYCLE as a generalization. Increasing k, the approximation of the algorithm satisfying $2k$CYCLE will be improved. A challenging subject is to establish the relation among approximation and k and speed of the algorithm. We conjecture that as long as k is constant, the fastest algorithm will work in linear time.

References

[1] D. Avis. "Worst Case Bounds for the Euclidean Matching Problem". In *International J. Comput. Math. Appl.*, 7:251–257, 1981.

[2] D. Avis. "A Survey of Heuristics for the Weighted Matching Problem". In *Networks*, 13:475–493, 1983.

[3] H. N. Gabow. "An Efficient Implementation of Edmonds' Algorithm for Maximum Matching on Graphs". *Journal of ACM*, 23:221–234, 1976.

[4] Z. Galil. "Efficient Algorithms for Finding Maximum Matching in Graphs". *Computing Surveys*, 18:23–38, 1987.

[5] J. E. Hopcroft and R. M. Karp. "$N^{5/2}$ algorithm for maximum matchings in bipartite graphs". *SIAM Journal on Computing*, 2:225–231, 1973.

[6] S. Micali and V. V. Vazirani. "An $O(\sqrt{|V|}|E|)$ Algorithm for Finding Maximum Matching in General Graphs". In *Proceedings of 21th Annual Symposium on the Foundations of Computer Science*, pages 17–27. IEEE, 1980.

[7] T. Nishizeki. "Lower Bounds on the Cardinality of the Maximum Matchings of Graphs". In *Proceedings of 9th Southeastern Conference on Combinatorics, Graph Theory, and Computing*, 1978.

[8] S. M. Venkatesan. "Approximation Algorithms for Weighted Matching". *Theoretical Computer Science*, 54:129–137, 1987.