Matching in general graphs

Visar Januzaj

Seminar: Integer programming and combinatorial optimization
Advisor: Ernst Althaus
February 18, 2003

Abstract. This paper is realised within the scope of the seminar "*Integer programming and combinatorial optimization*". The matching theory is one of the classical and most important topics in combinatorial theory and optimization [1]. We will discuss in this paper the matching problem in general graphs and we will focus on the Edmonds' *maximum cardinality matching algorithm*. We will handle the problem that the odd cycles, socalled blossoms, cause in general graphs by trying to find an augmenting path. An idea to **shrink** the blossoms will be shown, which was presented by Edmonds (1965) and ensures that an augmenting path can still be found if there is one.

1 Introduction

Comparing to the bipartite graphs finding an augmenting path is even more difficult for general graphs. Until 1965, where Edmonds in his pioneering work [9] solved the problem of the odd cycles, the socalled *blossoms*, by shrinking them, there were known only exponential algorithms for finding a maximum cardinality matching in general graphs. Even though since Berge's theorem (1957) it has been well known that for constructing a maximum matching, it suffices to search for augmenting paths. The reason was that one did not know how to treat the odd cycles in alternating paths. [2]
We will deal in this paper with the matching problem in general graphs, i.e. the bipartite-case will not be discussed. First, we will introduce some basic definitions and the Berge's theorem and then we will proceed with the Edmonds' algorithm.

2 Basic definitions

2.1 Matching

A ***matching*** of a graph $G = (V, E)$ is a subset of the edges $M \subseteq E$ such that no two edges in M touch a common vertex, i.e. $e \cap f = \emptyset$ for all $e, f \in M$ where $e \neq f$. We say a node v is ***covered*** by M, if there is an edge $e \in M$ that contains v. A ***perfect matching*** is a matching such that every node of the graph is covered, i.e. $|M| = \frac{|V|}{2}$. [3] In figure 1. we can see an illustration of a matching and a perfect matching, respectively. The dashed lines denote the

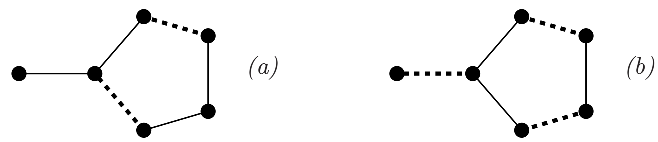

Fig. 1. Matching *(a)* and perfect matching *(b)*

matched edges.
A matching M is ***maximal*** if there exists no $e \in E \backslash M$ such that $M \cup \{e\}$ is a matching. A matching M is ***maximum*** if there exists no matching $M' \subseteq E$ of larger size. ***Maximum cardinality matching*** contains the maximum number of edges of a graph that form a matching. ***Maximum weighted matching*** is the maximal sum of the weights of the edges forming the matching.

2.2 Alternating and augmenting paths

Let $G = (V, E)$ be a graph (bipartite or not), and let M be some matching in G. We say that a path P is an ***M-alternating path*** if $E(P) \backslash M$ is a matching. An M-alternating path is ***M-augmenting*** if its endpoints are not covered by M, i.e. $|E(P) \backslash M| > |E(P) \cap M|$. [1] If P is an M-augmenting path then $|E(P) \backslash M| = |M| + 1$.
We can see that the nodes 1 and 6 in the graph in figure 2. are free, i.e. not covered by M, where $M = (2, 3), (4, 5)$. So, the path $P = 1, 2, 3, 4, 5, 6$ not only that it is M-alternating, but it is also M-augmenting.
It is easy to see that if we match the unmached edges and unmatch the matched

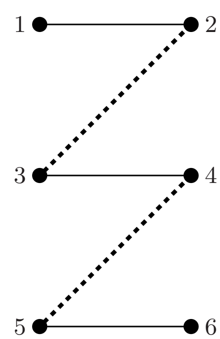

Fig. 2. An M-alternating path as well as M-augmenting

ones, we will get a new matching $M' = \{(1, 2), (3, 4), (5, 6)\}$, where $|M'| = |M| + 1$. This leads to the following:

Lemma 1: *Let M be a matching and P an M-augmenting path. Then $M \oplus E(P)$* [1] *is also a matching and $|M \oplus E(P)| = |M| + 1$ holds.* [4]

Proof. From the definition of the symmetric difference we get:
$|M \oplus E(P)| = |(M \cup E(P)) \backslash (M \cap E(P))|$
Considering $M = M_p \cup M_n$, where $M_p \subset E(P)$ and $M_n \cap E(P) = \emptyset$, we get the following:

$$\begin{aligned} |M \oplus E(P)| &= (|M \cup E(P)|) \backslash (|\underbrace{(M_p \cup M_n)}_{M} \cap E(P)|) \\ &= (|M \cup E(P)|) \backslash (|\underbrace{(M_p \cap E(P))}_{M_p} \cup \underbrace{(M_n \cap E(P))}_{\emptyset}|) \\ &= (|M \cup E(P)|) \backslash |M_p| \\ &= (|\underbrace{M \backslash M_p}_{M_n}|) \cup (|\underbrace{E(P) \backslash M_p}_{|M_p|+1}|) \\ &= |M| + 1. \end{aligned}$$

We know from the definition of the alternating path that $E(P) \backslash M_p$ is a matching and since P is an M_p-augmenting path then $|E(P) \backslash M_p| = |M_p| + 1$. So, we can conclude that $\underbrace{|M \oplus E(P)|}_{M'} > |M|$ and M' is a matching with $|M'| = |M| + 1$. $\square$

2.3 Berge's theorem

In 1957 Berge introduced one of the important theorem in order to find a maximum matching. He claimed that if a maximum matching has been reached then there is no augmenting path to find.

Theorem 1.: *Let G be a graph (bipartite or not) with some matching M. Then M is maximum if and only if there is no M-augmentig path.* [1]

Proof. Suppose there exists an M-augmenting path. From Lemma 1. we know that the symmetric difference $M \oplus E(P)$ is also a matching and has greater cardinality than M, so $|M \oplus E(P)| > |M|$, thus M is extendable $\Rightarrow$ M is not maximum.

Suppose conversely that M is not maximum. Let M' be a matching with $|M'| > |M|$. The symmetric difference $M' \oplus M$ yields a new graph where its connected components have the following attributes:

- $deg(v) \leq 2, \forall v \in V$ in $(V, M' \oplus M)$
- circles of even length
- paths (even and/or odd length)

[1] $\oplus$ denotes the symmetric difference, i.e. $a \oplus b = (a \cup b) \backslash (a \cap b)$

Let's say $C_1, C_2, \dots, C_n$ are these connected components in $(V, M' \oplus M)$ as mentioned above, then we have $M \oplus \underbrace{C_1 \oplus C_2 \oplus \cdots \oplus C_n}_{M' \oplus M} = M'$.

The C_i's containing (M-alternating) circles and paths with an even length are not of interest, because the number of edges which belong to M and M' is equal. We focus on the rest of the connected components containing odd paths. Only k-C_i's, $1 \leq k \leq n$, containing (M-augmenting) paths where the first and the last node are covered by M' make the cardinality of M greater, and that only for 1 each.

So, we can say that there exist $|M'| - |M|$ (node-disjoint) C_i's that are M-augmenting paths. □

2.4 Edmonds' shrinking idea

After Berge's theorem it took some time to find a way of treating the odd cycles in general graph, while trying to find an augmenting path. In 1965 Edmonds came up with an idea of shrinking these odd cycles, socalled blossoms. What is ment by *shrinking* is illustrated in the figure 3.

Definition 1.: *Let G be a graph and M a matching in G. A* ***blossom****, subgraph C, in G is an M-alternating path that forms a cycle of odd length with $|M \cap E(C)| = \frac{|V(C)|-1}{2}$. A node in C which is incident to two unmatched edges, i.e. not covered by $M \cap E(C)$, is called the* ***base*** *of blossom C.* [1]

The next Lemma is the base of Edmonds' cardinality matching algorithm, where he claims that, if there is no augmenting path to find after shrinking, then there is no augmenting path in the original graph either.

Lemma 2.: *Let G be a graph, M a matching in G, C a blossom in G (with respect to M). Suppose there is an M-augmenting v-r-path Q of even length from a node v not covered by M to the base r of C, where $E(Q) \cap E(C) = \emptyset$. Let G' and M' result from G and M by shrinking ($V(C)$ to a single node. Then M is a maximum matching in G if and only if M' is a maximum in G'.*

Proof: Suppose M is not a maximum matching in G. Let $N := M \oplus Q$ be

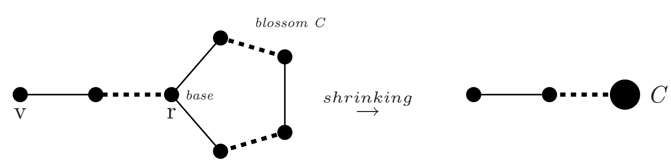

Fig. 3. Shrinking a blossom

a matching not covering the base r of the blossom C, $r \notin V(N)$, and $|N|{=}|M|$. Using *Berge*'s theorem we know that there exists an N-augmenting path P in G. Let $P_1, P_2, \ldots, P_n$ denote the nodes in P, where $|P| = n$. We call $P_1, \ldots, P_n$ ***C-free*** if they do not touch the blossom C. P has the following attributes:

– either, both endpoints P_1 and P_n are C-free, i.e. $P \cap V(C) = \emptyset$,
– or, only one of the endpoints P_1 is C-free, i.e. $P_i \in V(C), 1 < i \leq n$.

Let M', N' and P' result after shrinking C in G to the single node C_b in G'. We get $|N'| = |M'|$ and $P' = P'_1, \ldots, P'_k$. Depending on which of the above mentioned attributes P has, we say: if P_n is C-free then $P'_k = P_n$ $(k = n)$, otherwise $P'_k = P_i$ where P_i is the first node belonging to C, so $P'_k = C_b$. If the first case occures then we can easily say that P' is an N'-augmenting path. Does the second case occure then we have to show that $P'_1, \ldots, P'_k$ is an N'-augmenting path. We know that N doesn't cover the base of C, so N' doesn't cover C_b either and because of $P'_k = C_b$ follows that both endpoints of P' are not covered by N'. Thus, P' is an N'-augmenting path what means that N' is not a maximum matching in G'. Because of $|N'| = |M'|$ follows that M' is not a maximum matching in G' either.
Suppose conversely M' is not a maximum matching in G'. Let N' be a matching with greater cardinality than M', $|N'|{>}|M'|$, where $|N'|{=}|N_0|$ and N_0 is the correspondent matching in G covering at most one node of C. From the definition of the blossom we say that N_0 can be extended by $k = \frac{|V(C)|-1}{2}$ edges, so we get:

$$|N| = |N_0| + k = |N'| + k > |M'| + k = |M|$$

It is easy to see from the equation above that there exists a matching N in G which is greater than M, $|N| > |M|$, so M is not a maximum matching in G either. $\square$

One may ask why it is important to have an alternating path of even length from a node not covered by the matching to the base of the blossom (see Lemma 2.). The answer is that we might destroy the only existing augmenting path while shrinking. Such a situation is illustrated in the figure bellow.

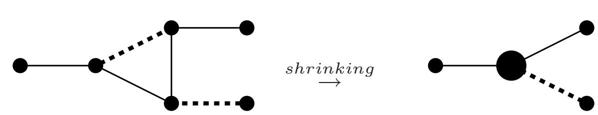

Fig. 4. Destroying the only existing augmenting path

3 Edmonds' cardinality matching algorithm

The idea of shrinking the found blossom(s), finding an augmenting path if there is one and then expand the shrunken blossom(s) is the gist of the Edmonds' matching algorithm (Lemma 2.). While trying to find an augmenting path an alternating forest will be built up. We define in the following the alternating forest:

Definition 2.: *Given a graph G and a matching M in G. An* ***alternating forest*** *with respect to M in G is a forest F in G with the following properties:*

i) *$V(F)$ contains all the nodes not covered by M,*
ii) *each connected component of F contains exactly one node not covered by M, its* ***root****,*
iii) *all inner*[2] *nodes have degree 2 in F and*
iv) *for any $v \in V(F)$, the* ***unique path*** *$P(v)$ from v to the root of the connected component containing v is M-alternating. [1] and [5]*

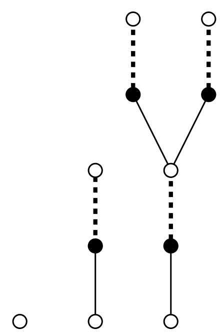

Fig. 5. An alternating forest

How an alternating forest looks like is illustrated in figure 5. The black filled nodes are inner, the unfilled (white) nodes are outer and, as we know, the dashed edges belong to the matching.
Now we can make the last step and introduce Edmonds' algorithm. However, since there are a lot of improvements of this algorithm we present here a description of it, i.e. the core. Although this algorithm has been implemented in various ways, e.g. the chosen data structure, *the core* remains the same.
The algorithm starts with an empty matching M and with the set of nodes not covered by M. It builds a forest when constructing a matching. At any stage of the algorithm we consider a neighbour y of an outer node x.

[2] we call a node $v \in V(F)$ with an even resp. odd distance to the root of the connected component containing v, an **outer** resp. **inner** node. Roots are outer nodes.

In the following we show the three important and interesting cases, corresponding to three operations (”grow”, ”augment” and ”shrink”), on which the algorithm is based:

1. $y \notin V(F)$. Then the forest will **grow** when we add $\{x, y\}$ and the matching edge covering y.
2. y is an outer node in a different connected component of F. Then we **augment** M along $P(x) \cup \{x, y\} \cup P(y)$.
3. y is an outer node in the same connected component of F (with root q). Let r be the first node of $P(x)$ (starting at x) also belonging to $P(y)$. (r can be one of x, y.) If r is not a root, it must have degree at least 3. So r is an outer node. Therefore $C := P(x)_{[x,r]} \cup \{x, y\} \cup P(y)_{[y,r]}$ is a blossom with at least three nodes. We **shrink** C. [1]

If none of the cases applies, all the neighbours of outer nodes are inner. We claim that M is maximum.
Since the algorithm is based on the already proved theorems and lemmas we will not prove the algorithm explicitly.

4 Conclusion

We have presented in this paper the main idea of the Edmonds' cardinality matching algorithm. Even though we did not present an implementation of the algorithm, we did show the three important operations, namely, *grow*, *augment* and *shrink*. We also discussed Berge's theorem and Edmonds' shrinking lemma, on which the algorithm is based.
The first polynomial running time of the algorithm was $O(n^4)$, where n is the number of nodes in a given graph. An $O(n^3)$-implementation of the algorithm has been presented by Gabow, Lovsz and Plummer etc. The currently best known algorithm for the cardinality matching problem has running time of $O(\sqrt{n}m)$, presented by Micali and Vazirani in 1980.
The matching algorithm can be extended to the weighted case, which appears to be one of the ”hardest” combinatorial optimization problems that can be solved in polynomial time. The first $O(n^3)$-implementation of Edmonds' algorithm for the minimum weight perfect matching problem and the theoretically best running time $O(mn + n^2\log n)$ has been obtained by Gabow in 1973 resp. 1999. [1]
The general literature for this paper was [1], but it is also strongly related to [2]-[9].
In the following we will show some chosen text from the Jack Edmonds' original paper [9] about the matching algorithm.

4.1 Jack Edmonds' paths, trees and flowers

> *First, what I present is a conseptual description of an algorithm and not a particular formalized algorithm or ”code”.*
> *For practical purposes computational details are vital. However, my purpose is only to show as attractively as I can that there is an efficient algorithm.* Jack Edmonds

A *matching* in G is a subset of its edges such that no two meet the same vertex. We describe an efficient algorithm for finding in a given graph a matching of maximum cardinality. This problem was posed and partly solved by C.Berge.
Maximum matching is an aspect of a topic, treated in books on graph theory, which has developed during the last 75 years through the work of about a dozen authors. In particular, W.T.Tutte (1947) characterized graphs which do not contain a *perfect* matching, or *1-factor* as he calls it - that is a set of edges with exactly one member meeting each vertex. His theorem prompted attempts at finding an efficient construction for perfect matching.
Berge proposed searching for augmenting paths as an algorithm for maximum matching. In fact, he proposed to trace out an alternating path from an exposed vertex until it must stop and, then, if it is not augmenting, to back up a little and try again, thereby exhausting possibilities.
The algorithm which is being constructed is efficient because it does not require tracing many various combinations of the same edge in order to find an augmenting path or to determine that there are none.
When *flowers* arise we "shrink" the blossoms, and so if an augmenting path arises later, it will be in a "reduced" graph.
An upper bound on the order of difficulty of the matching algorithm in n^4, where n is the number of vertices in the graph. The algorithm consists of "growing" a number of trees in the graph - at most n - until they augment or become Hungarian[3]. A tree is growing by branching from a vertex in a tree to an edge-vertex pair not yet in the tree - at most n times. Such a branching may give rise to a back-tracing through at most n edge-vertex pairs in the tree in order to relabel some of them as forming a blossom or an augmenting path.
A possible alternative to actually shrinking is some method for tracing through the internal structure of a pseudovertex. Witzgall and Zahn (1965) have designed a variation of the algorithm which does that. Their result is attractive and deceptively non-trivial.

References

1. B.Korte and J.Vygen *Computational Optimization*. Springer
2. Norbert Blum *Maximum Matching in Nonbipartite Graphs Without Explicit Consideration of Blossoms*. Universität Bonn, 1999
3. Angelika Steger *Diskrete Strukturen, band 1*. Springer-Verlag, 2001
4. Ernst W. Mayer *Effiziente Algorithmen und Datenstrukturen*. TU-München, 1998
5. Reiner Schrader *Graphentheorie*. Universität zu Köln, 2003
6. Sven O. Krumke *Integer programming*. Technische Universität Kaiserslautern, 2003
7. Robert Tarjan *COS 423: Theory of Algorithms*. Princeton University, 2002
8. Andrew J. Mason *Speaker Matching, A Third Professional Project*. University of Auckland, 1987
9. Jack Edmonds *Paths, trees, and flowers*. Canad. J. Math., 17 (1965), 449-467

[3] A *Hungarian tree* H in a graph G is an alternating tree whose outer vertices are joined by edges of G only to its inner vertices