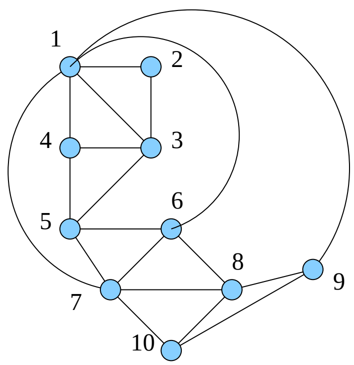
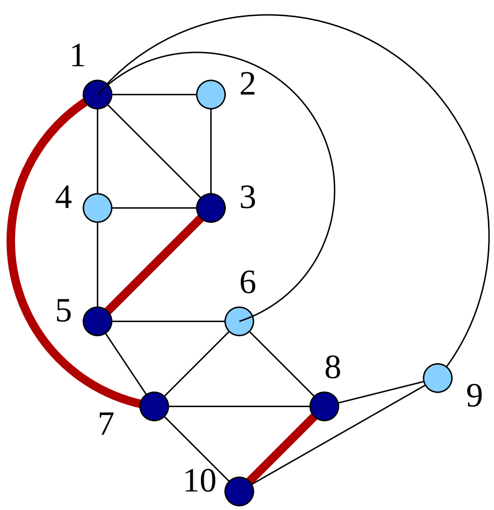
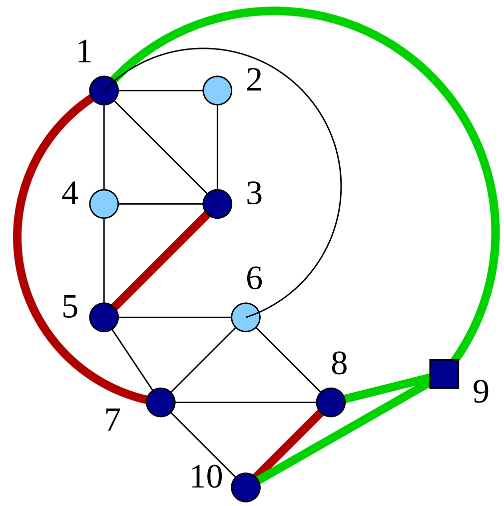
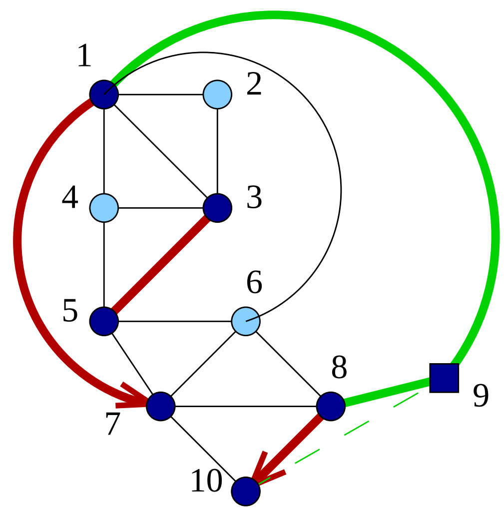
Edmonds’ algorithm

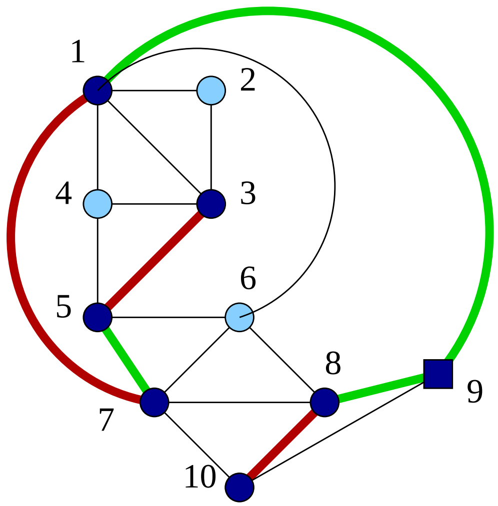
There are many many details about the algorithm that we have not considered. As usual the best is to see this with an example. Consider the following graph and try Edmonds’ algorithm on it.



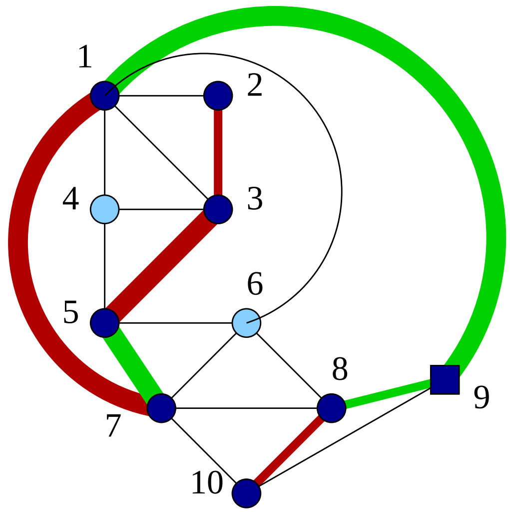




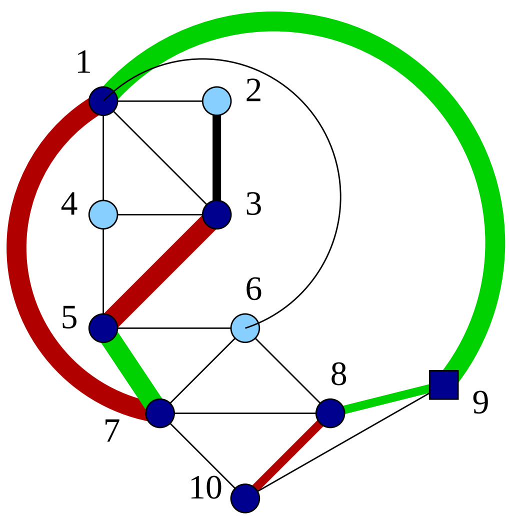




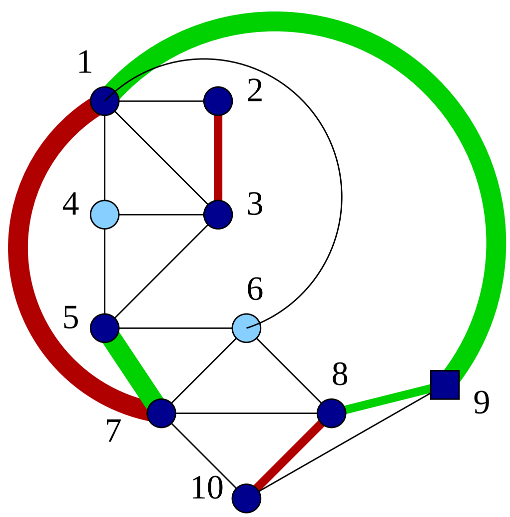
5



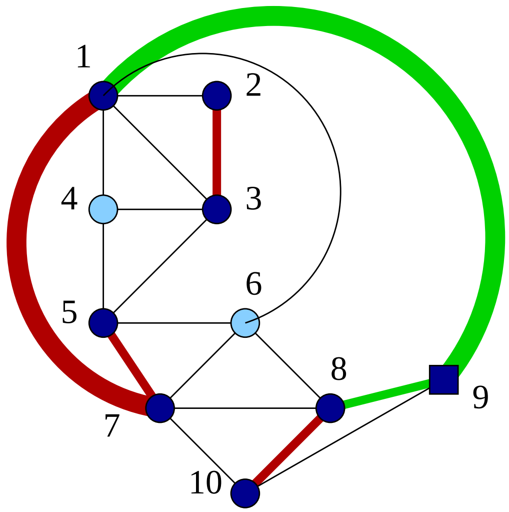
7



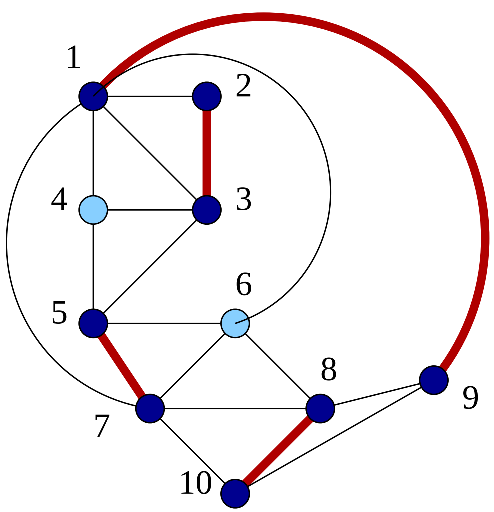
6



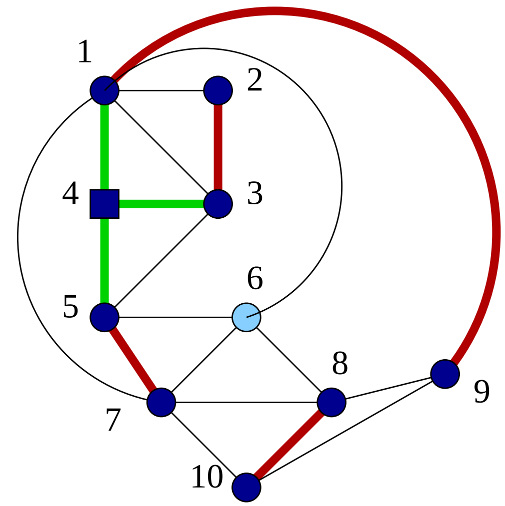
8



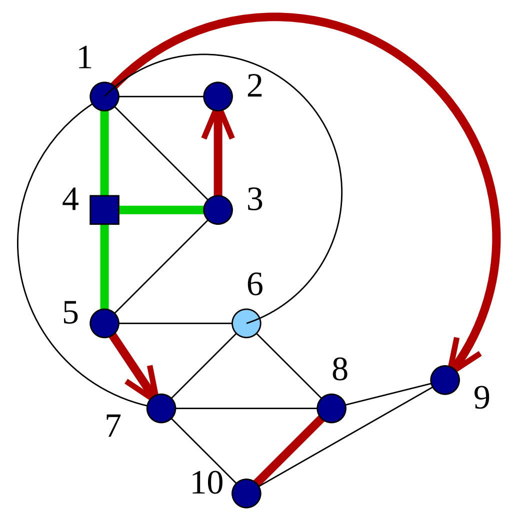
9



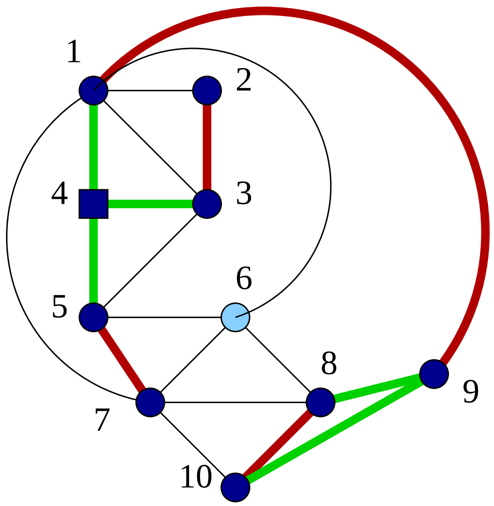
10

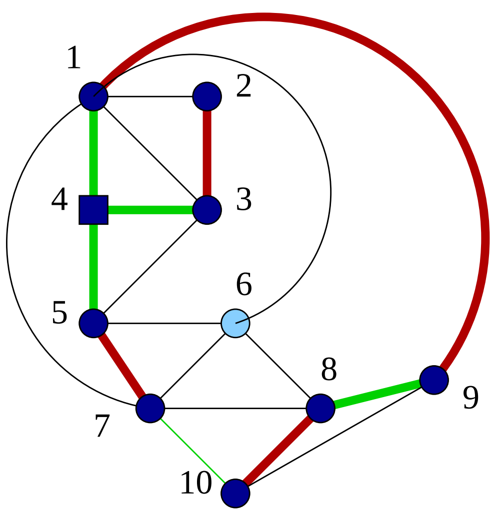


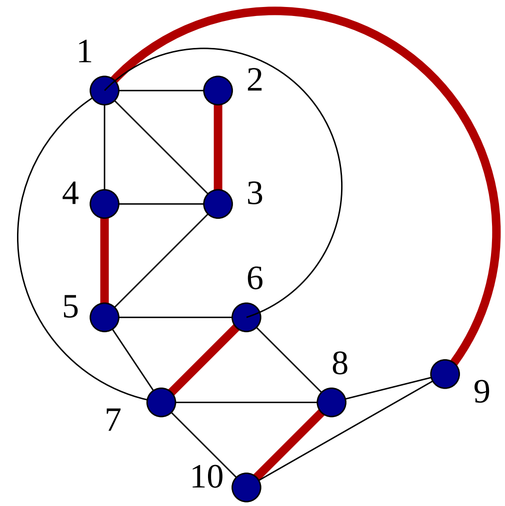
11



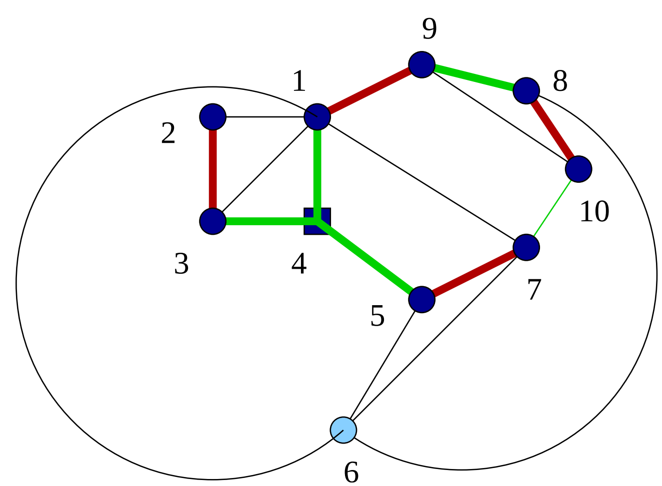
12



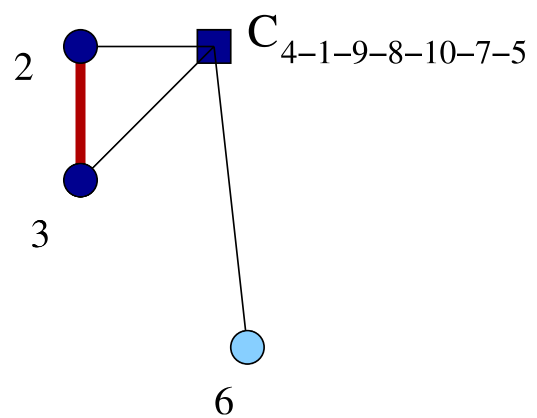




Alternative approach



Reduced graph



In summary we can always do one of the following:

- enlarge F (if there is an outer vertex adjacent to $y \in V(M) \setminus V(F)$),
- enlarge M (if two outer vertices in different components are adjacent),
- decrease $|V(G)|$ (if we find a “blossom” or a “flower”), or
- stop

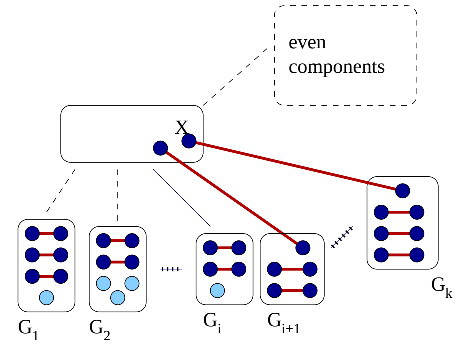
What can we say about the “quality” of M when we stop?

Correctness

The *deficiency* of a graph G is def$(G) = |V(G)| - 2\nu(G)$ (i.e. the number of vertices left uncovered by a maximum cardinality matching).

$c_o(G)$ = the number of connected components of odd size in G.

Theorem. (Berge formula) def$(G) = \max\{c_o(G \setminus X) - |X| : X \subseteq V(G)\}$.



(def$(G) \geq \max_{X \subseteq V(G)}\{c_o(G \setminus X) - |X|\}$) Let X be any set of vertices in G and let M be a maximum matching in G. Let $G_1, G_2, \ldots, G_i, \ldots, G_k$ be the odd cardinality connected components in $G \setminus X$. Without loss of generality assume the first i of these have at least one vertex not covered by M. This has two consequences:

1. def$(G) \geq i$;
2. $|X| \geq k - i$ (because for each of the $k - i$ odd components of $G \setminus X$ whose vertices are completely covered by M there must be[a] an edge of M with one endpoint in X and one in G_j).

The saught inequality follows.

[a] Each of these components has an odd number of vertices so on of its vertices must be covered by an edge of M that joins the given component to X.

$(\text{def}(G) \leq \max_{X \subseteq V(G)}\{c_o(G \setminus X) - |X|\})$ We will show that there exists an X for which equality holds. We prove this by induction on the number of vertices of G.

BASE: trivial.

STEP: Two cases need to be considered.

Case 1. $\exists v\ \nu(G \setminus v) < \nu(G)$. By definition of deficiency $\text{def}(G \setminus v) = n - 1 - 2(\nu(G) - 1) = \text{def}(G) + 1$. By the induction hypothesis there exists a set of vertices X' in $G \setminus v$ such that $\text{def}(G \setminus v) = c_0(G \setminus v \setminus X') - |X'|$. Let $X = X' \cup \{v\}$. Using the two last equations we have

$$\text{def}(G) + \cancel{1} = c_o(G \setminus X) - |X| + \cancel{1}.$$

Case 2. $\forall v\ \nu(G \setminus v) = \nu(G)$. ... Well, in this case we will prove that G has a matching that misses exactly one vertex from each of G's connected components ... but to prove this we need another result first!

A theorem of Gallai

A graph G is said to be *factor-critical* (or *hypomatchable*) if $G - v$ has a perfect matching for every $v \in v(G)$. It is clear that if G is factor-critical then $\nu(G - v) = \nu(G)$ for each $v \in V(G)$.

Gallai proved that for connected graphs the converse holds as well.

Theorem. If G is connected and $\nu(G - v) = \nu(G)$ for each $v \in V(G)$ then G is factor-critical.

Back to Edmonds' algorithm

At each iteration, once the forest F is completed three cases arise.

Case 1 There exist two adjacent outer vertices in different components

AUGMENTING PATH!

Case 2 There exist two adjacent outer vertices in the same component.

SWITCH & CYCLE SHRINKING!

Case 3 All neighbours of the outer vertices are inner vertices.

THE MATCHING IS MAXIMUM!

(Exercise: convince yourself that no other case arises).

Close-up on Case 3

Suppose that F contains m inner vertices and n outer vertices.

Clearly $|S| = n - m$.

Furthermore if we delete all the inner vertices of F from G, the remaining graph will contain all the outer vertices of F as isolated points (i.e. odd components!).

Hence $\text{def}(G) \geq n - m = |S|$.

But M misses exactly $|S|$ vertices, and so it must be a maximum matching!