

C0759 Term Project

Cutting Plane Method for Matching and
Its Application in Solving Uncapacitated
b-Matching via Data Reduction

By Steely Glint

Tian QIAO (Editor), t2qiao@uwaterloo.ca
Department of Applied Mathematics
University of Waterloo, Waterloo, Ontario,
Canada N2L 3G1

Yinuo LIU (Coder), y652liu@uwaterloo.ca
Computational Mathematics
University of Waterloo, Waterloo, Ontario,
Canada N2L 3G1

Uncapacitated b-matching

First, what is a matching?

In graph $G=(V,E)$, a matching M is a subset of edges in E that each vertex in V only meet M at most once.

If each vertex in V meets M exactly once, then it is called a perfect matching

The indicator for matching:

$$x_e = \begin{cases} 1, & \text{if } e \text{ is in } M \\ 0, & \text{otherwise} \end{cases}$$



The Integer Programming for Perfect Matching

$$\begin{aligned} \min \quad & c_e^T x_e \\ & x(\delta(v)) = 1, \forall v \in V \\ & x = \{0, 1\}; \end{aligned}$$

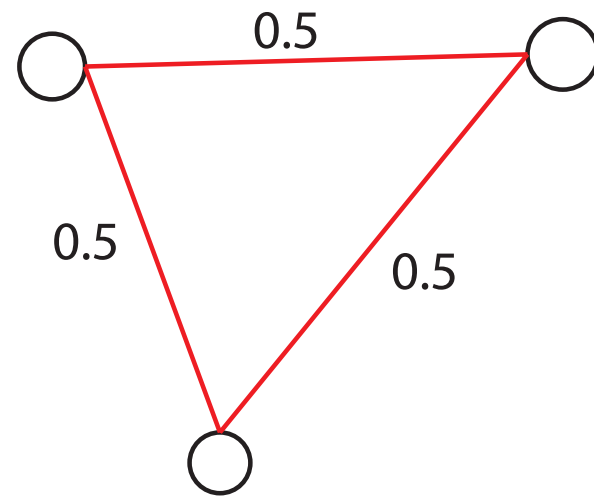
The IP is generally hard to solve, so we take the relaxation of it

The LP Relaxation

$$\begin{aligned} \text{(P)} \quad & \min c_e^T x_e \\ & x(\delta(v)) = 1, \forall v \in V \\ & x \geq 0; \end{aligned}$$

However, solving (P) will not give us what we want

An counter example:



In order to avoid
this, we need cutting
planes:

Blossom Inequality

$$x(\gamma(S)) \leq \frac{|S| - 1}{2}$$

$\forall S \subset V, S$ is odd

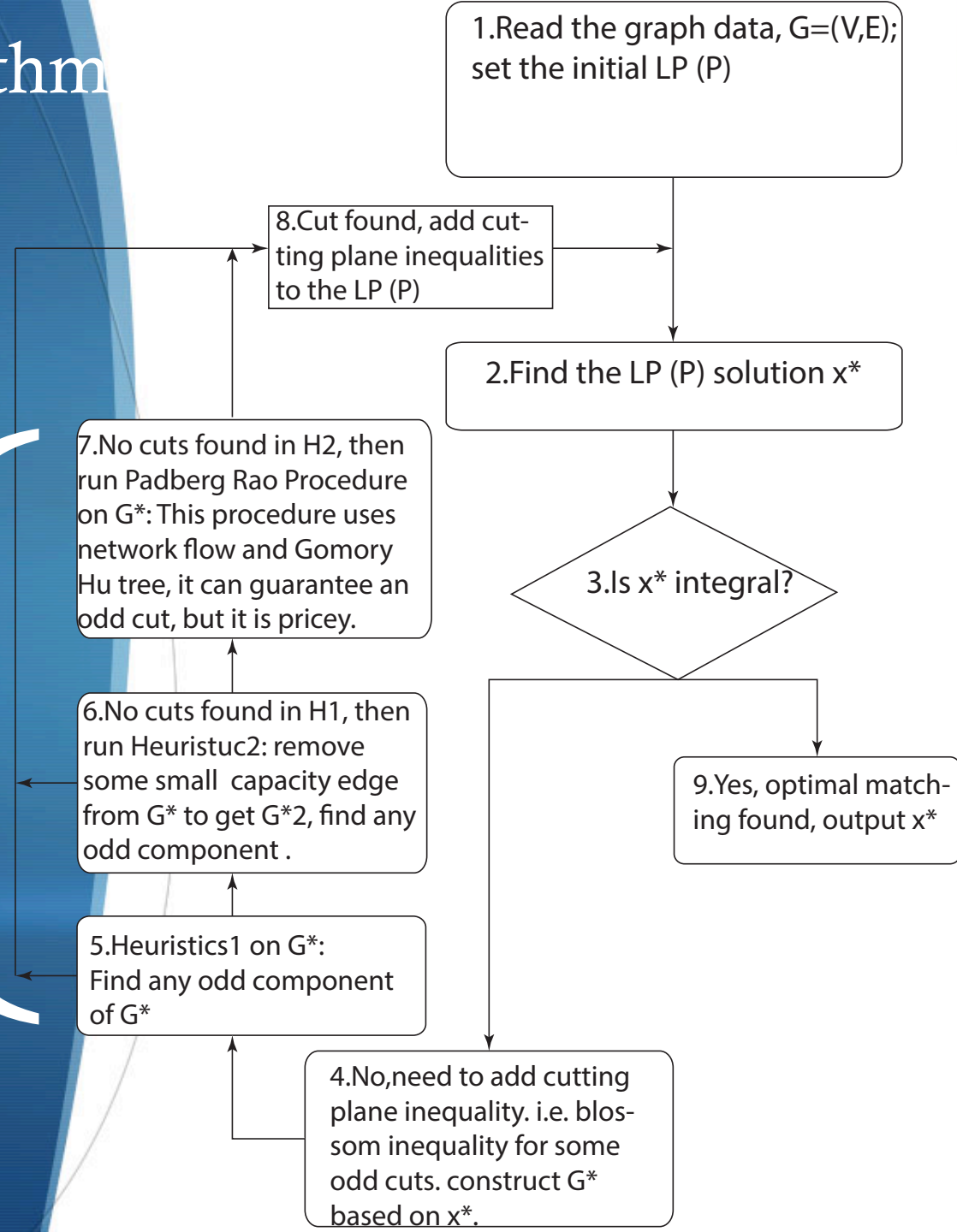
Or equivalently

$$x(\delta(S)) \geq 1$$

$\forall S \subset V, S$ is odd

Cutting Plane Algorithm for Perfect Matching

This is for finding odd cuts that violate the blossom inequality, add them as cutting plane.



Heuristics

The LP graph G^* is a graph with same vertices as G , and non-zero LP solution edges of G . The capacity of edges of G^* is the LP solution .

Heuristics1 can find disconnected components of G^* . These components have cut capacity zero , and the odd components will definitely violate the blossom inequalities.

Heuristics2 is based on Heuristic1. But remove some weak edge i.e. edges with capacity less than 0.3. Heuristics2 will give some odd cut with very small cut capacity.

Death of Heuristics

Heuristics are cheap. Only $O(n)$ complexity.

However, they could **fail**!

When there are lots of Combs.

When there are no odd componenets.

What should we do if they **fail**?

We have an ultimate weapon of massive destruction :

Padberg and Rao Procedure

But before it, we need another definition

Gomory-Hu Tree

Let $\text{MAXFLOW}(u, v, G^*)$ be the maximum flow between u, v in G^*

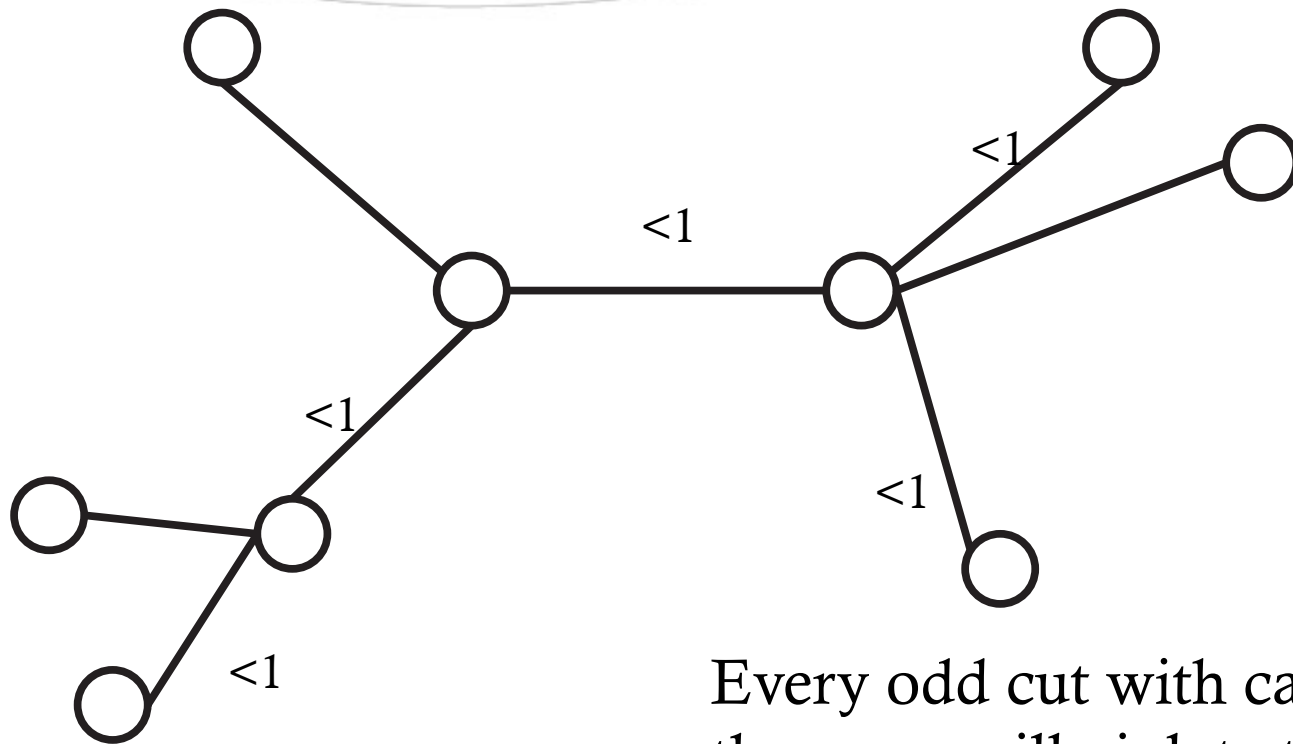
Let $\text{PATH}(u, v, T)$ be the unique path between u, v in tree T

Let $\text{GH}(G^*)$ be the Gomory-Hu tree of G^* which is defined as:

$V(\text{GH}(G^*)) = V(G^*), \forall u, v \in V(G^*),$
if $uv \in \text{GH}(G^),$ then $w_{uv} = \text{MAXFLOW}(u, v, G^*);$*
if $uv \notin \text{GH}(G^),$*
then $\min\{w_{ij} : ij \in \text{PATH}(u, v, \text{GH}(G^))\} = \text{MAXFLOW}(u, v, G^*);$*
let $mn = \{mn \in \text{GH}(G^) : w_{mn} = \min\{w_{ij} : ij \in \text{PATH}(u, v, \text{GH}(G^*))\}\},$*
then disconnect mn in $\text{GH}(G^)$ gives two components and the components*
*gives a minimum capacity $u - v$ cut in G^**

Padberg and Rao $O(V^3 E^2)$

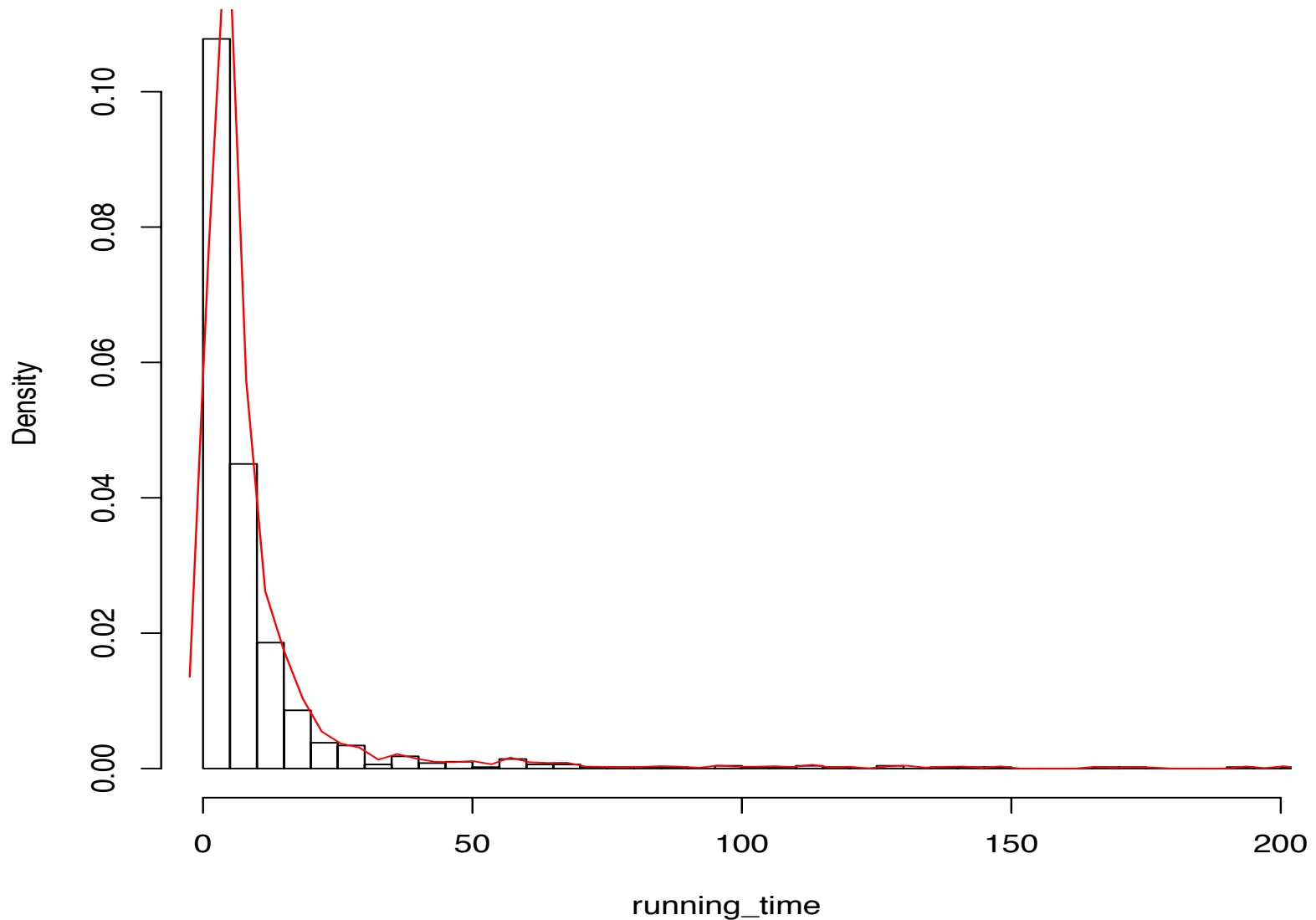
GH(G^*):



Every odd cut with capacity less than one will violate the blossom inequality.

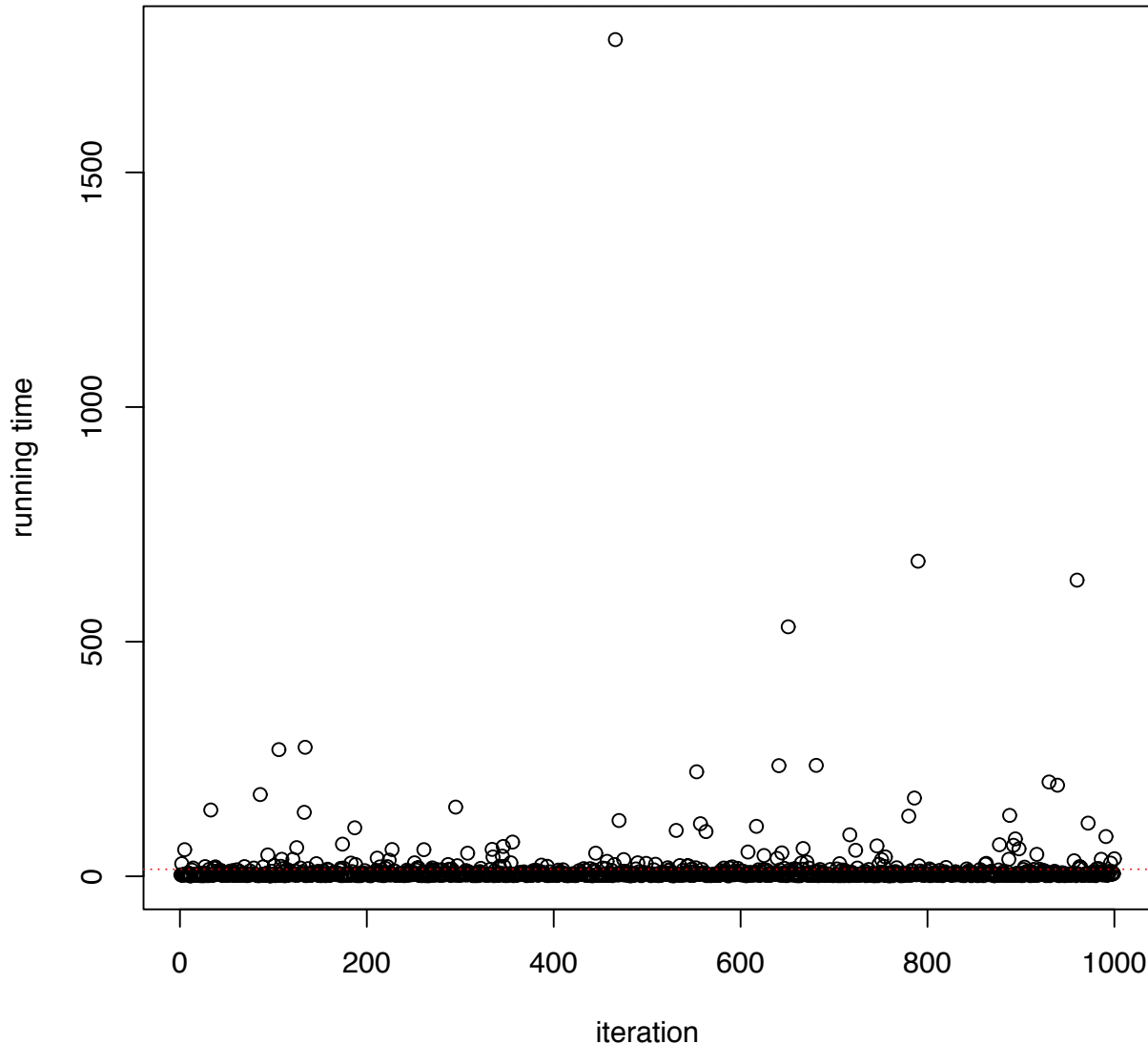
Computational Test on 1000 Random Complete Graph K1000

Histogram of running time



Computational Test on 1000 Random Complete Graph K1000

scatter plot of running time



Median : 4.68

Mean: 14.92s

3rd Quantile:
9.344s

Peak: 1783s

Uncapacitied b-matching

Just like the perfect matching, the b-matching is a matching M_b such that

$$|M_b \cap v| = b_v, \forall v \in V$$

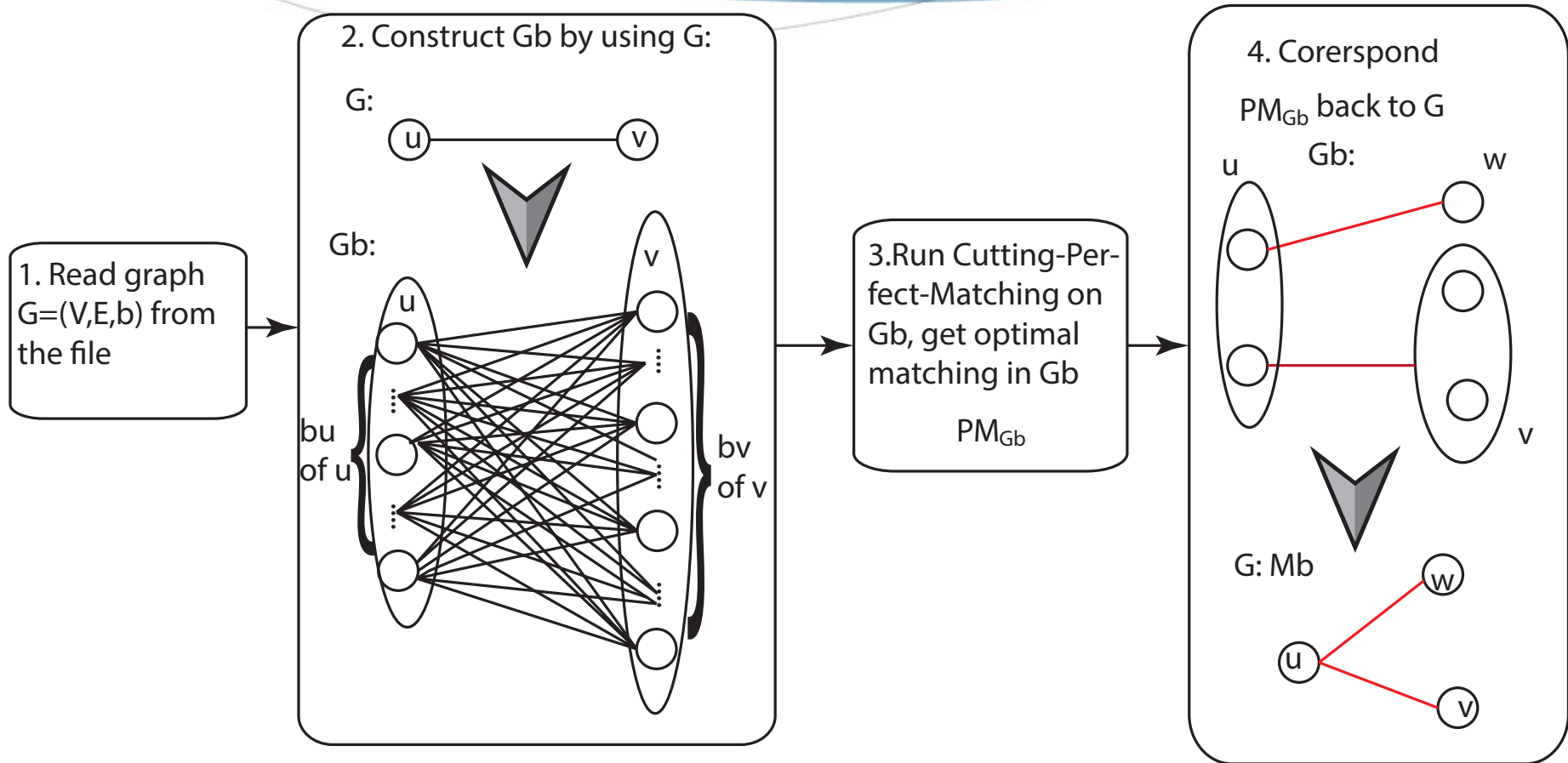
Where b is a vector with $\dim(b) = |V|$

If each edge in M_b can take arbitrary value, then this is an uncapacitied b-matching.

One can reduce b-matching to perfect-matching.



Graph Reduction



Cons: Increase the usage of storage.

Pros: Strong polynomial time

Looking forward

- ◆ Use blossom algorithm instead of cutting plane. More stable.
- ◆ Modify the graph further to solve capacited b-matching

Thank you!