UEFA Champions League Draw 2024

During the UEFA Champions League 2024, UEFA aims to distribute the top 36 teams into 4 pots and create a match schedule where each team faces 2 teams from each pot, with the condition that two teams from the same country cannot compete against each other.

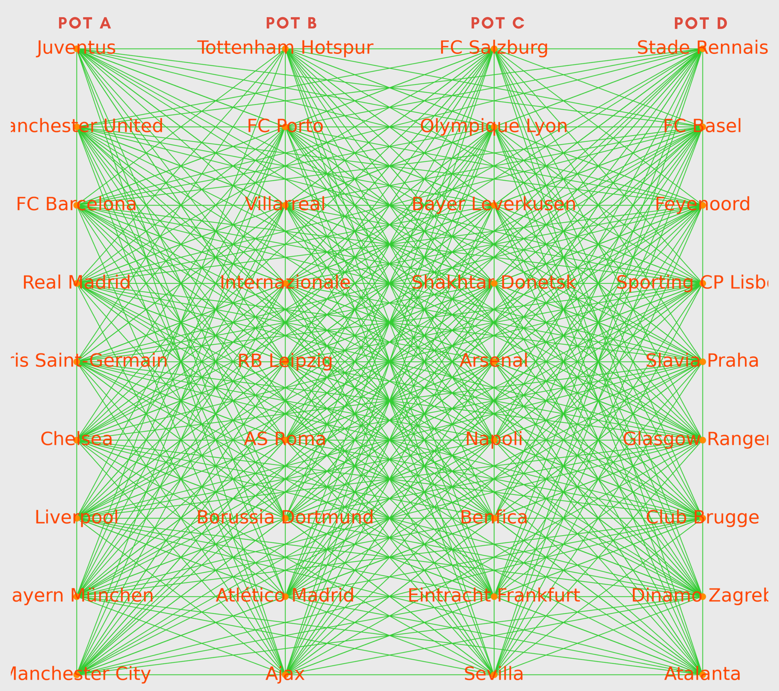
GENERAL PROBLEM

One way to approach this problem is graphically. We consider the graph formed by the 4 pots, where the vertices represent the teams, and two vertices are connected by an edge if a match is possible (subject to the constraint of different countries). We are interested in the general case formulated as follows:

> **OBJECTIVE**
>
> **INPUT :** $G = (V, E)$ *graph, a* partition $V_1, \ldots, V_k$ of V, a an integer *d*
>
> **OUTPUT :** "yes" If there exists a subset F of E such that every vertex has d neighbors in each V_i

The problem **is separable into two distinct problems**, depending on whether we are looking for d neighbors between two different pots or within the same pot.

During the presentation, we will illustrate our results applied to the Champions League problem, i.e., the case of d = 2 in the following graph:



I. FIRST DECOMPOSITION

The first problem is as follows: for each pair of distinct integers i, j from $[k]$: decide whether there exists a subset of edges in the bipartite graph $(V_i \cup V_j, E_{ij})$, giving each vertex a degree of d, where E_{ij} is the subset of E with one endpoint in V_i and the other endpoint in V_j

> **INPUT :** $G = (U \cup V, E)$ bipartite graph, *d* an integer
>
> **OUTPUT :** Find a subset $F \subseteq E$ such that, $\forall v \in V \; deg_F(v) = d.$

This is solved by a maximum flow problem.
Consider:

$$G = (U \cup V \cup \{s, t\}, E \cup \{(s, u), (v, t) | u \in U, v \in V\})$$

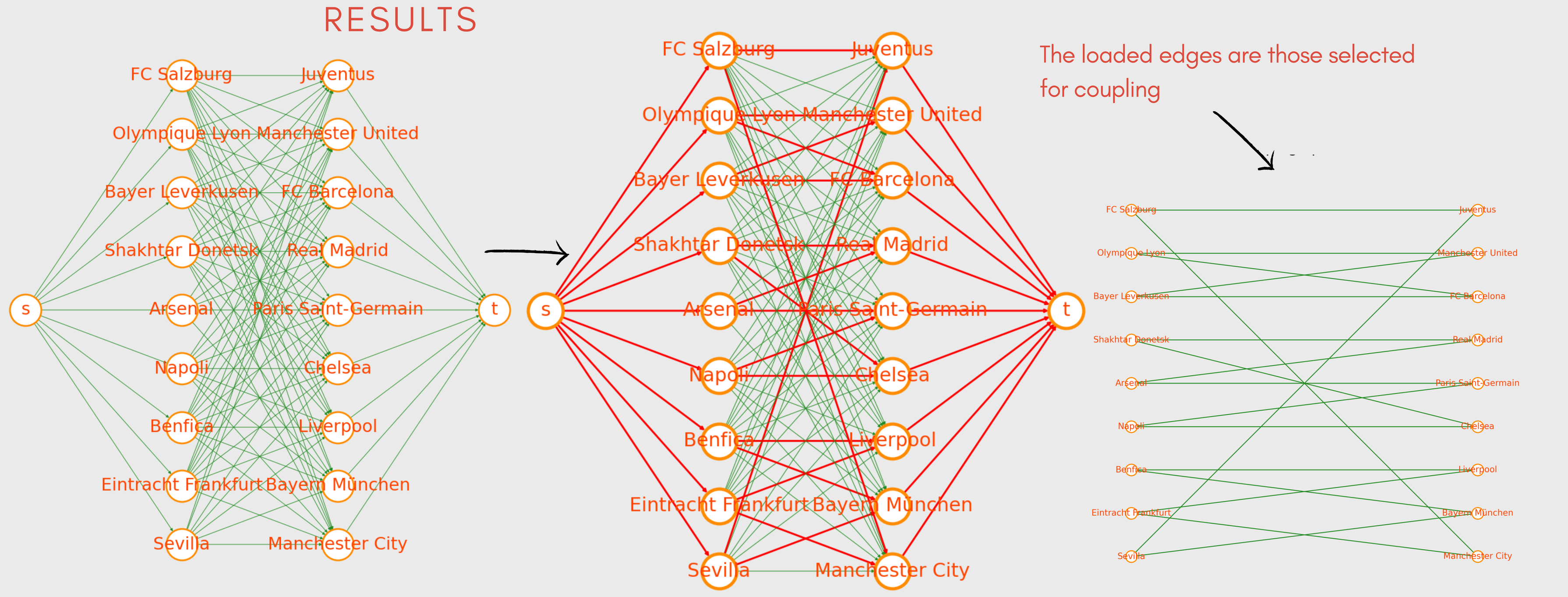
with capacities :

- $\forall (u, v) \in U \times V \quad c(u, v) = 1$
- $\forall u \in U \quad c(s, u) = d$
- $\forall v \in V \quad c(v, t) = d$

Algorithm used : Edmonds-karp, with a complexity of $O(|V||E|^2)$

RESULTS

The loaded edges are those selected for coupling



GRAPH CONSTRUCTION S-T FLOT — MAXIMUM S-T FLOT — FOUND MATCHING

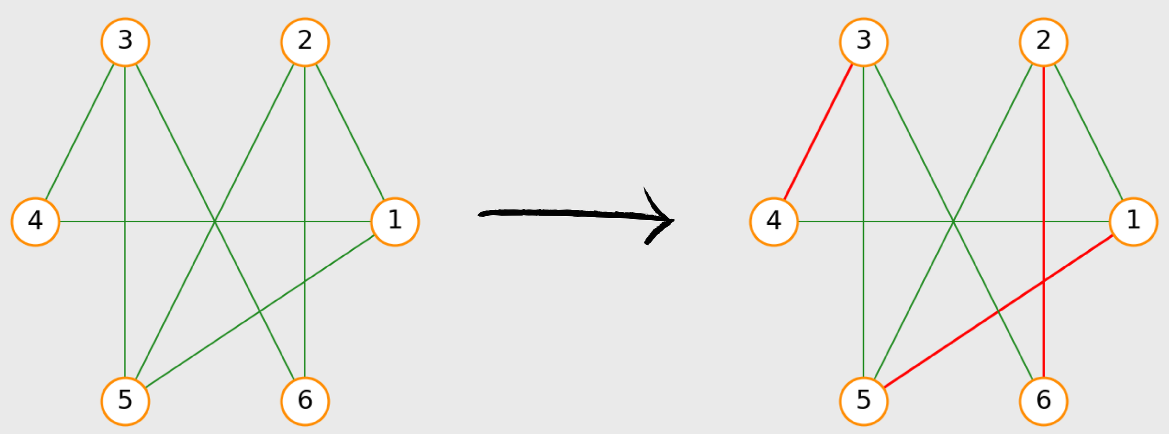
II. SECOND DECOMPOSITION

The second problem is as follows: for each integer i in [k], decide whether there exists a subset of edges in the "induced" subgraph $G[V_i] = (V_i, E_i)$, giving each vertex a degree of d, where E_i is the subset of E with both endpoints in V_i. This can be solved in polynomial time by a "*simple d-matchings*" algorithm

> **INPUT :** $G = (V, E)$ arbitrary graph, integer *d*
>
> **OUTPUT :** Find a *simple-d-matching*

Case $d = 1$

In the case where $d = 1$, the problem consists of finding a perfect matching in a graph. ***Edmonds' algorithm for matchings*** (also known as the blossom algorithm), commonly referred to as the algorithm of flowers and petals, is an algorithm for constructing maximal matchings in graphs. The algorithm has a time complexity of $O(|E||V|^2)$



A perfect or "simple" matching is a matching M in the graph such that every vertex in the graph is incident to exactly one edge of M.

Case arbitrary d

Construction of a new graph

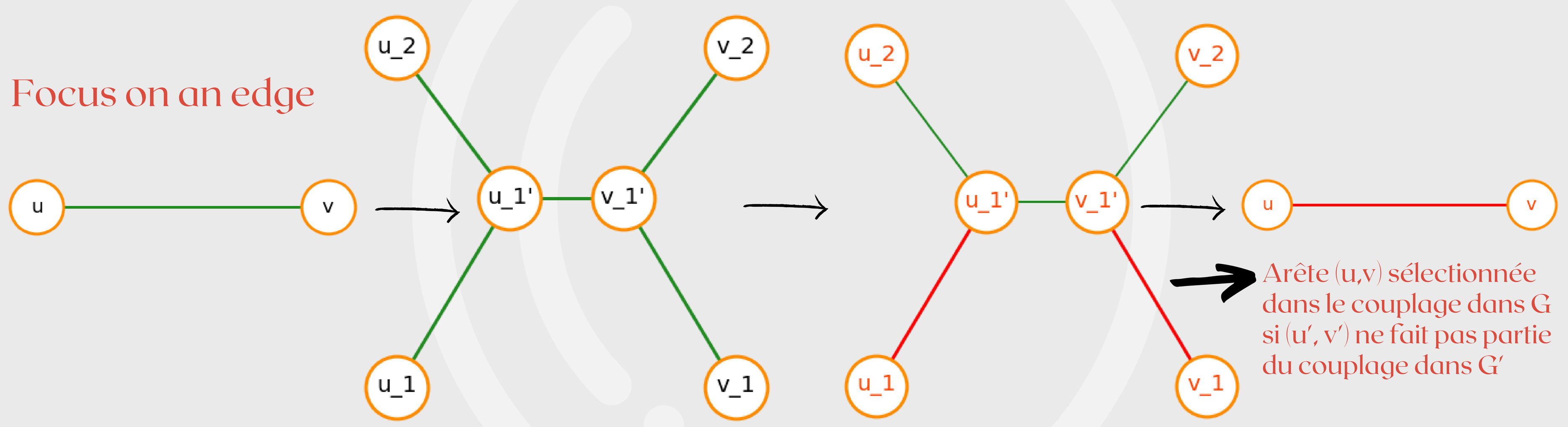
We consider the graph constructed from $G = (V, E)$ as follows:

- **Replace** each vertex u with d copies $u_1, \ldots, u_d$
- For each edge (u, v) (which is removed): create two vertices u′ and v′ connected by an edge, add d edges between the u_i and u′, add d edges between the v_i and v′

Let the new graph be $G' = (d|V| + 2|E|, (2d+1)|E|)$.

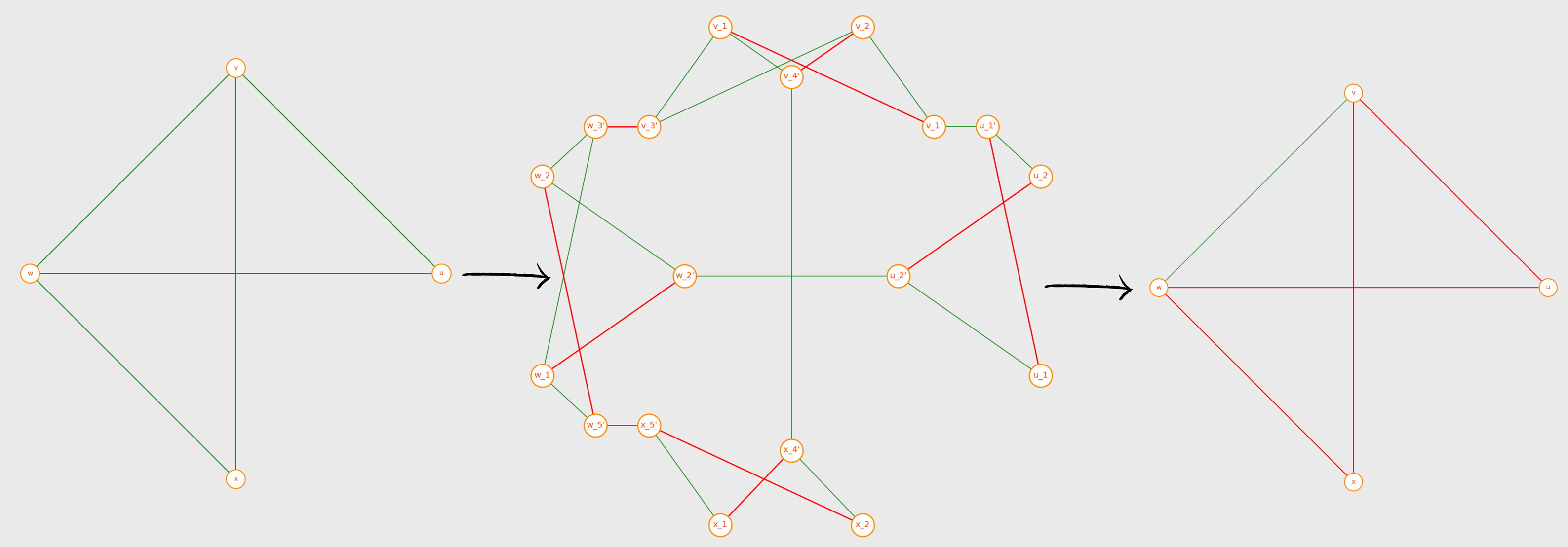
> **Lemma :** There exists a simple-d-matching in G if and only if there exists a perfect matching in the graph G′.
>
> *Furthermore, it is possible to construct a simple-d-matching in the graph G if one has a perfect matching in the graph G′, and vice versa.*

Focus on an edge

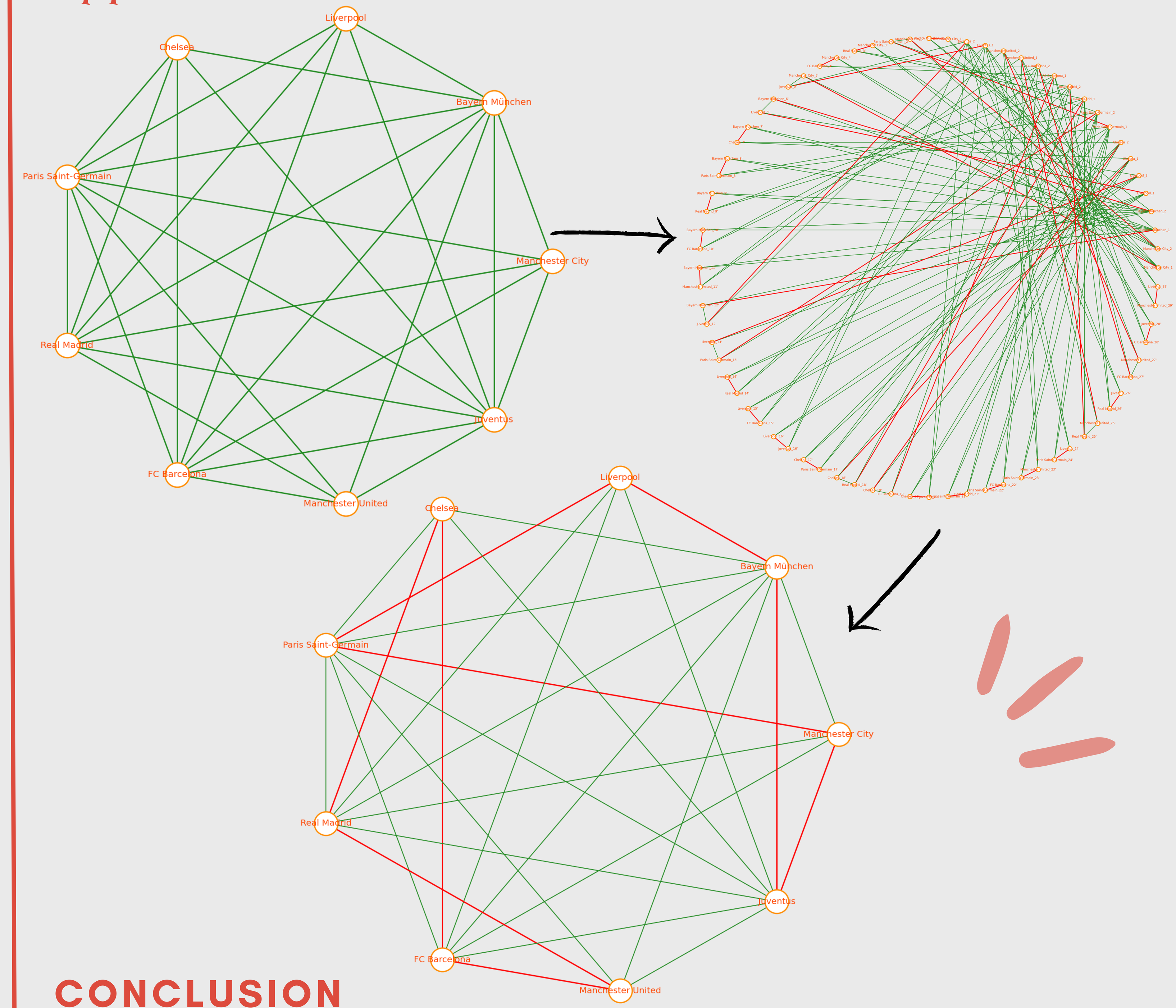


Remark: It is possible to generalize the result in the case where d(v) is a function that associates the number of neighbors in the graph to each node. The result remains true by making d(v) copies.

Illustration in a small graph



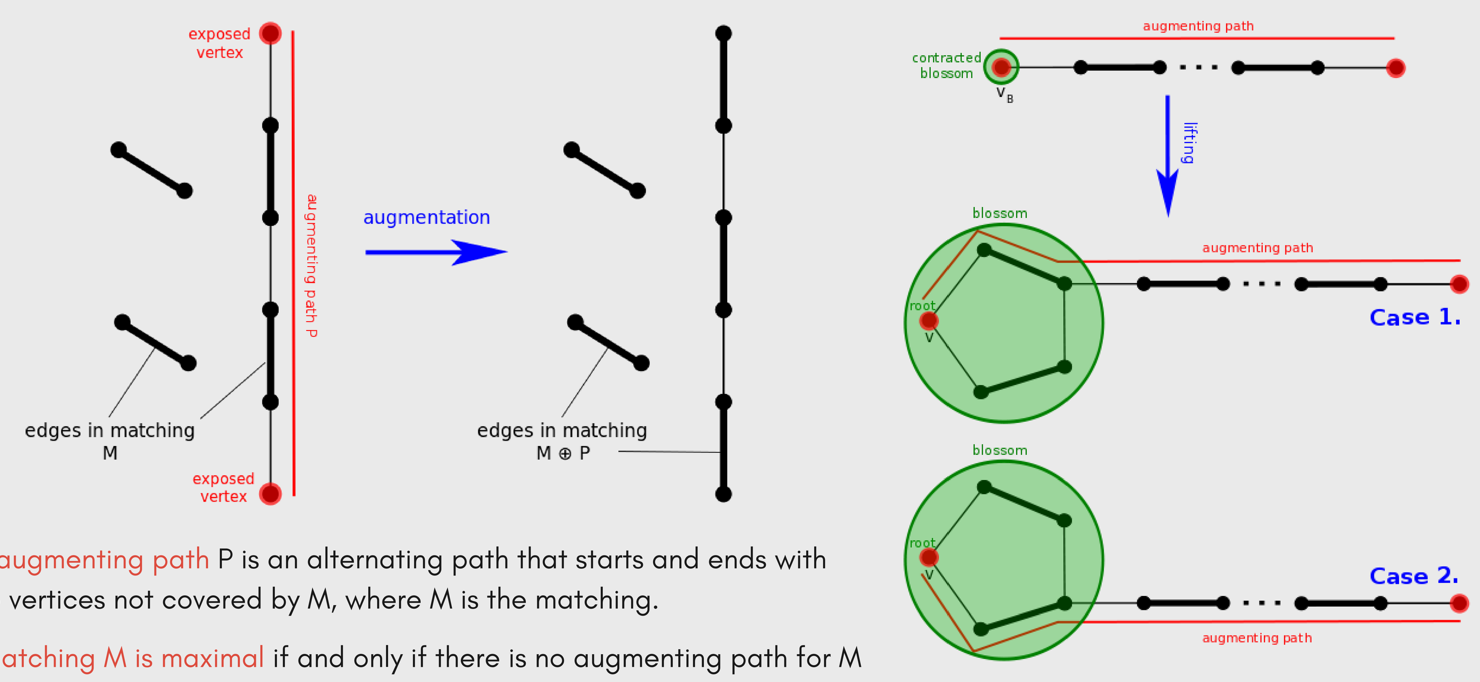
Application to the case d = 2



CONCLUSION

> We have a solution if and only if the answer is "yes" for the previous $\frac{n(n-1)}{2} + n = \frac{n(n+1)}{2}$ tests. In the new Champions League model, we find that it is possible to find such a pairing, namely, we obtain "yes" in the case $d=2$ in the graph explained earlier.
>
> To establish a **match schedule** in accordance with UEFA rules, you need to add an edge to the coupling if you can complete it with a **simple-2-matching**.

ANNEXE Explanation of Edmonds' algorithm



An augmenting path P is an alternating path that starts and ends with two vertices not covered by M, where M is the matching.

A matching M is maximal if and only if there is no augmenting path for M in G.

source : https://fr.wikipedia.org/wiki/Algorithme_d%27Edmonds_pour_les_couplages