We propose a constraint programming (CP) approach to the popular matching problem. We show that one can use the Global Cardinality Constraint to encode the problem even in cases that involve ties in the ordinal preferences of the applicants. We also tackle a more general case where additional copies of posts can be made in order to find a popular matching.

**Keywords**
Preferences, Matching problems, Constraint Programming

1. INTRODUCTION

Matching problems involving preferences are well known in real world applications such as the assignment of junior doctors to hospital [5], students to campus housing [6, 9], patients to donors [8] and so on. The notion of popular matching was introduced by Gardenfors [4], but it has its origins from as far back as 1785 with the notion of Condorcet winner. Stable matching problems, first introduced by Gale and Shapley in 1962 [3], have been studied deeply over the past decade. Different formulations of these problems have been proposed, distinguishing between one-sided matching, e.g. house allocation, and two-sided matching, e.g. the stable marriage problem.

Surprisingly, popular matching has never been studied in the context of CP. In this paper we study this problem and propose the first CP formulation of it. We consider two cases of the problem of popular matching - instances with and without ties in the preference lists - and show that one can elegantly encode these problems using the global cardinality global constraint [7]. Moreover we present a number of new graph properties for the extended version with copies, the FixingCopies problem, very efficient to tackle the problem.

We consider the problem of matching a set of applicants to a set of posts, where each applicant has a preference list, ranking a nonempty subset of posts in order of preference, possibly involving ties [1]. An instance of the popular matching problem is a bipartite graph $G = (A \cup P, E)$, where $A$ is the set of applicants, $P$ is the set of posts, and $E$ is a set of edges. If $(a, p) \in E$, and $(a, p') \in E_j$ with $i < j$ then we say that $a$ prefers $p$ to $p'$. If $i = j$ we say that $a$ is indifferent between $p$ and $p'$. This ordering of posts adjacent to $a$ is called $a$’s preference list. If applicants can be indifferent between posts we say that preference lists contain ties. Let $M$ be a matching of $G$, a vertex $a \in A \cup P$ is either unmatched in $M$, or matched to some vertex denoted by $M(a)$ (i.e. $(u, M(u)) \in M$). An applicant $a$ prefers a matching $M'$ to $M$ if $a$ is matched in $M'$ and unmatched in $M$, or a prefers $M'(a)$ to $M(a)$. $M'$ is said more popular than $M$ if the number of applicants that prefer $M'$ to $M$ exceeds the number of applicants that prefer $M$ to $M'$.

**Definition 1.1** (Popular Matching). A matching $M$ is popular if and only if there is no matching $M'$ that is more popular than $M$.

**Example 1.1.** $A = \{a_1, a_2, a_3\}$, $P = \{p_1, p_2, p_3\}$ and each applicant prefers $p_1$ to $p_2$ and $p_2$ to $p_3$. This instance does not admit a popular matching.

2. POPULAR MATCHING IN CP

Similar to [1], we assume that every applicant $a_i \in A$, has in its preference list an extra unique post $l_i$, called the last resort, that is worst than any other post in $P$. In this way every applicant is guaranteed to be matched. We use one integer variable $x_i$ per applicant $a_i$. The domain of each $x_i$ represents all posts that are neighbours of $a_i$, denoted by $N(a_i)$, plus the unique last post $l_i$. The domain is initialised as $D(x_i) = \{j | p_j \in N(a_i)\} \cup \{\{P\} + i\}$. Assigning a value $k$ to $x_i$ is interpreted as $a_i$ is matched to post $p_k$ if $k \leq |P|$, and to $l_i$ otherwise. Let $[x_1, \ldots, x_n]$ be a sequence of integer variables and $\Delta = \bigcup_{i=1}^n D(x_i)$. Let $lb$ and $ub$ be two mappings on integers such that $lb(j) \leq ub(j)$ for all $j$. The Global Cardinality Constraint GCC [7] restricts the occurrences of any value $j \in \Delta$ in the sequence $[x_1, \ldots, x_n]$ to be in the interval $[lb(j), ub(j)]$.

**Definition 2.1.** GCC($lb, ub, [x_1, \ldots, x_n]$) : $\bigwedge_{j \in \Delta} lb(j) \leq |\{i | x_i = j\}| \leq ub(j)$. 
Next we will distinguish between preference lists without ties and with ties.

2.1 Preferences Without Ties

For each applicant $a_i$, we denote by $f(a_i)$ the best post in its preference list. A post $p_j \in P$ is called an $f$-post if $\exists a_i \in A$ such that $f(a_i) = p_j$. We denote by $s(a_i)$ the best post for $a_i$ that is not an $f$-post.

**Lemma 2.1.** [1] A matching $M$ is popular iff:

- every $f$-post is matched,
- for each applicant $a_i$, $M(a_i) \in \{f(a_i), s(a_i)\}$.

Using Lemma 2.1 we can model the popular matching problem using one GCC problem. First we define $x^*$ which is a popular matching.

2.2 Preferences With Ties

In this case $f(a_i)$ is defined as the set of top choices for applicant $a_i$. However the definition of $s(a_i)$ is no longer the same. Indeed it may now contain any number of surplus $f$-posts.

We will use the Gallai-Edmonds decomposition [2] for bipartite graphs. Let $G_1 = (A \cup P, E_1)$ where $E_1 \subseteq E$ is the subset of edges corresponding to top choices. Let $M$ be a maximum cardinality matching in $G_1$. The three set of vertices: even (resp. odd) is the set of vertices having an even (resp. odd) alternating path (w.r.t $M$) in $G_1$ from an unmatched vertex; and unreachable is the set of vertices that are not in even $\cup$ odd. We denote by $E$, $O$, $U$ the sets of even, odd, and unreachable vertices, resp.

**Lemma 2.2 (Gallai-Edmonds decomposition).** Let $G_1$, $O$, $U$ be the vertices sets defined by $G_1$ and $M$ above. Then:

a) $E$, $O$, and $U$ are a partition of $A \cup P$, and any maximum matching in $G_1$ leads to exactly the same sets $E$, $O$, and $U$.

b) Every node in $O$ (resp. $U$) is matched to a node in $E$ (resp. $U$), and $|M| = |O| + |U|/2$.

c) No maximum matching of $G_1$ contains an edge between two nodes in $O$, a node in $O$ and a node in $U$, or between a node in $E$ and a node in $U$.

We define $s(a)$ the set of top-ranked posts in $a$’s preference list that are even in $G_1$.

**Lemma 2.3.** [1] A matching $M$ is popular iff:

- $M \cap E_1$ is a maximum matching of $G_1$,

- For each applicant $a_i$, $M(a_i) \in \{f(a_i), s(a_i)\}$.

We can model the popular matching problem with ties using one GCC constraint. First, the domain is pruned with $D(x_i) \leftarrow f(a_i) \cup s(a_i), \forall i \in [1,|A|]$. Next, by using the properties of $E$, $O$, $U$ from Lemma 2.2 we can apply several preprocessing steps to our model (note that a last resort can be only a member of $E$):

A. By definition, every vertex in $U$ is matched to another vertex in $U$. Let $\Psi = \{j | p_j \in U\}$ and $\Omega = \{i | a_i \in U\}$ be the two sets of vertices in $U$. This property leads to these preprocessing steps:

- $\forall i \in \Omega, D(x_i) \leftarrow D(x_i) \cap \Psi$,

- $\forall j \in \Psi, \forall i \in [1,|A|] \setminus \Omega, D(x_i) \leftarrow D(x_i) \setminus \{j\}$.

B. By definition, every vertex in $O$ is matched to a vertex in $E$. Let $\Theta = \{j | p_j \in E \lor j \in E\}$ and $\Upsilon = \{i | a_i \in E\}$ be the two sets of vertices in $E$. Let $\Phi = \{k | a_k \in \Omega\}$ and $\Lambda = \{l | p_l \in O\}$ be the two sets of vertices in $O$. This property leads to these preprocessing steps:

- $\forall k \in \Phi, D(x_k) \leftarrow D(x_k) \cap \Theta$,

- $\forall l \in \Lambda, \forall i \in [1,|A|] \setminus \Upsilon, D(x_i) \leftarrow D(x_i) \setminus \{l\}$.

The values of $lb(j)$, and $ub(j)$ are defined as follows: a) $lb(j) = 1$, for all $j$ such that $p_j \in O \cup U$, otherwise $lb(j) = 0$; b) $ub(j) = 0$ for all $j$ such that $\forall a_i \in A, f(a_i) \neq p_j$ and $s(a_i) \neq p_j$, otherwise $ub(j) = 1$.

**Theorem 2.2.** $GCC(lb, ub, [x_1, \ldots, x_{|A|}])$ is satisfiable iff $M$ is a popular matching with ties.

3. FIXING COPIES PROBLEM

It is assumed that the Fixing Copies problem does not admit a popular matching for the basic instance $I$.

**Instance:** Given a graph $G = (A \cup P, E)$ and a list $(c_1, \ldots, c_{|P|})$ of upper bounds on the number of copies possible for each post.

**Question:** Does there exist an $(\chi_1, \ldots, \chi_{|P|})$ such that for each $i \in \{1, \ldots, |P|\}$, having $c_i$ copies of the $i$-th post, where $1 \leq \chi_i \leq c_i$, enables the resulting graph to admit a popular matching?

The FixingCopies problem is known to be $\mathcal{NP}$-complete, and it remains $\mathcal{NP}$-complete even for the 1-or-2 copies problem. In this case the upper bounds $c_i$ are either 1 or 2. From the basic instance, the problem involves adding, or not, copies of each post with the aim of obtaining a new instance that admits a popular matching. Thus, this problem can be divided into two parts:

a) the decision of the number of copies for each post that defines an instance $I'$ of the popular matching problem;

b) solving the popular matching problem given by the new instance $I'$ using the CP model with the Global Cardinality Constraint.

The difficulty of this problem is to find which post should be copied first in order to find a solution quickly. Using interesting graph properties about any copy of a post in the sets $E, O, U$, we can use the automaton in Figure 1 to improve the branching strategies based on the $E, O, U$ labelling.

![Figure 1: Every post follows this automaton after a copy.](image)

**Pruning:** For any instance $I$, we denote by $F^I$ the set of posts that are not $f$-posts or $s$-posts.
**Lemma 3.1.** Let $\mathcal{I}$ and $\mathcal{I}^*$ be two instances where $\mathcal{I}^*$ is built from $\mathcal{I}$ by copying some posts. Any post in $\mathcal{F}^*$ is in $\mathcal{F}^\mathcal{I}^*$.

**Theorem 3.1.** Let $\mathcal{I}$ be an instance without a popular matching, and let $p_i \in \mathcal{F}^\mathcal{I}$. Every instance with a popular matching, obtained from $\mathcal{I}$ by copying some posts, remains popular even with the original number of copies of $p_i$ from $\mathcal{I}$.

In Figure 2(a), the instance $\mathcal{I}$ doesn’t admit a popular matching, but by adding some copies of some post we can obtain a solution. In Figure 2(b), we notice the effect of a post copy in the graph, in black line the f-posts, in bold the maximum matching selected, in dash line the s-posts selected.

**Figure 2:** Illustration of the branching strategy for the FixingCopies problem.

### 4. EXPERIMENTAL RESULTS

We implemented the CP models in Numberjack using the Mistral solver and the standard approach implemented in C++. We generated large random instances where the number of applicants is equal to the number of posts. The results are showing that the CP model is as competitive as the standard algorithm.

<table>
<thead>
<tr>
<th># Applicants</th>
<th>8.5k</th>
<th>9k</th>
<th>9.5k</th>
<th>10k</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Without ties</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard</td>
<td>144</td>
<td>160</td>
<td>177</td>
<td>195</td>
</tr>
<tr>
<td>Mistral</td>
<td>152</td>
<td>168</td>
<td>191</td>
<td>215</td>
</tr>
<tr>
<td><strong>With ties</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard</td>
<td>153</td>
<td>170</td>
<td>162</td>
<td>211</td>
</tr>
<tr>
<td>Mistral</td>
<td>144</td>
<td>162</td>
<td>181</td>
<td>204</td>
</tr>
</tbody>
</table>

For the FixingCopies problem, we generated the instances using a preferential attachment mechanism. The generated graph is used to compute an initial ranking for the posts. Next, the preference lists are added one by one and each post is ranked using a probability to follow the initial ranking. We compare different exploration strategies from the six permutations to rank the three labels $\mathcal{E}, \mathcal{O}, \mathcal{U}$. We also use a lexicographical exploration. In Figure 4 we can see 4 branching strategies using the pruning rule. The first observation from Figure 4 is that all enumeration strategies starting from posts in $\mathcal{O}$ is the best strategy to branch on.

### 5. CONCLUSION AND FUTURE RESEARCH

We proposed the first CP formulation for the popular matching problem which can handle cases in which there are ties in the applicants’ preference lists. We introduced new graph properties for this extended case, and from these results we introduced different enumeration methods as well as a pruning rule. These experiments showed essentially that the hierarchical ranking strategy (based on our graph study) is extremely beneficial. As part of our future work we will focus on solving more general problems embedding popular matching.

### 6. REFERENCES