A CP-Based Approach to Popular Matching

Sorina Chica, Mohamed Siala, Gilles Simonin, Barry O’Sullivan
Insight Centre for Data Analytics, University College Cork, Ireland
(sorina.chica, mohamed.siala, gilles.simonin, barry.osullivan)@insight-centre.org

Introduction
The popular matching problem involves finding a matching $M$ between applicants and posts such that there exists no matching $M'$ where more applicants prefer $M'$ to $M$.

Let $G = (A \cup P, E)$ be an instance of the popular matching problem, where
- $A$ are the set of applicants
- $P$ the set of posts
- $E$ is set of edges.

Goal
Propose a Constraint Programming (CP) model to solve the popular matching problem.

Definition A matching $M$ is popular iff there is no matching $M'$ that is more popular than $M$.

Figure 1: An instance that doesn’t admit a popular matching.

Previous work
Stable Matching Problems (SM) and its variants in a CP context have been exhaustively studied. Some efficient algorithms for solving popular matching problems, e.g., [Abraham et al.].

Description
- One integer variable $x$ per applicant $a$.
- Domain of $a$ is $D(a) = \{(a, p_i) \in E \} \cup \{l\}$.

For each applicant $a$, we denote by
- $f(a)$ the best post in its preference list, called $f$-post.
- $l$ the last-resort post $\notin P$, called $l$-post.

A post $p_j \in P$ is called an $f$-post if $\forall a \in A$ such that $f(a) = p_j$.

Preferences without ties
For each applicant $a$, we denote by $s(a)$ the best choice for $a_j$ that is not an $f$-post, called $s$-post.

Lemma [Abraham et al.] A matching $M$ is popular iff the following conditions hold:
- Every $f$-post is matched, and
- For each applicant $a$, $M(a) \in \{f(a), s(a)\}$.

GCC model
The domain of every variable $x_i$ is reduced to be exactly $\{f(a), s(a)\} \forall i \in [1, |A|]$.

Next, we define $lb(i)$ and $ub(i)$ as follows:
- $lb(i) = 1$ if $p_i$ is an $f$-post, else $lb(i) = 0$.
- $ub(i) = 0$ if $p_i \in A$, $f(a) \neq p_i$ and $s(a) \neq p_i$, else $ub(i) = 1$.

Theorem [Chica et al.]
If $GCC(lb, ub, \{x_1, \ldots, x_n\})$ is satisfiable iff $M$ is a popular matching.

Preferences with ties
The definition of $s(a)$ is no longer the same. In this case it may contain a number of surplus $f$-posts.

Let $G_1 = (A \cup P, E_1)$ be a bipartite graph of top choices.

Let $M$ be a maximum cardinality matching in $G_1$.

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

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

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

The $s$-posts are defined as the top choice(s) for $a_j$ that is not in $E$.

Figure 2: An example with ties, a maximum matching in bold and the $E$, $O$, $U$ labeling.

Lemma [Abraham et al.] A matching $M$ is popular iff the following conditions hold:
- $M \cap E_1$ is a maximum matching of $G_1$, and
- For each applicant $a$, $M(a) \in \{f(a), s(a)\}$.

GCC model
The domain of every variable $x_i$ is reduced to be exactly $\{f(a) \cup s(a)\} \forall i \in [1, |A|]$.

Using Gallai-Edmonds decomposition as preprocessing rules we will prune the domain of the variables.

We can define $lb(i)$ and $ub(i)$ as follows:
- $lb(i) = 1 \forall j \in E \times [1, |I|]$.
- $ub(i) = 0 \forall j \in O \times [1, |I|]$.
- $s(a) \neq p_i$.

Theorem [Chica et al.]
If $GCC(lb, ub, \{x_1, \ldots, x_n\})$ is satisfiable iff $M$ is a popular matching with ties.

Popular Matching with Copies
Consider the case where extra copies of posts are allowed in order to find a solution. This problem is called the FixingCopies problem and it is well known to be $NP$-complete. Assume that the FixingCopies problem does not have a solution for the basic instance $I$.

This problem can be divided into two parts:
- The decision of the number of copies for each post that defines an instance $I'$ of the popular matching problem.
- Solving the popular matching problem given by $I'$.

The difficulty of this problem is to find which post should be copied first in order to find a solution quickly.

An Automaton for the Posts
Using our new graph results about any copy of a post in the sets $E$, $O$, $U$ we obtain the following automaton:

Figure 3: The effect of a post copy in black the $E$-posts, in red the maximum matching selected, in green the $s$-posts selected.

Pruning
For any instance $I$, we denote by $F^2$ the set of posts that are not $f$-posts or $s$-posts.

Lemma Let $I$ and $I^*$ be two instances where $I^*$ is built from $I$ by copying some posts. Any post in $F^2$ is in $F^2$.

Theorem Let $I$ be an instance without a popular matching, and let $p_i \in F^2$. Any instance with a popular matching, obtained from $I$ by copying some posts, remains popular even with the original number of copies of $p_i$ from $I$.

Results

Table 1: Popular matching: Running time of standard algorithm vs. CP model.

<table>
<thead>
<tr>
<th># Applicants</th>
<th>50k</th>
<th>10k</th>
<th>25k</th>
<th>50k</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>144</td>
<td>160</td>
<td>177</td>
<td>219</td>
</tr>
<tr>
<td>Mistral</td>
<td>152</td>
<td>168</td>
<td>215</td>
<td>215</td>
</tr>
</tbody>
</table>

Without ties

With ties

Figure 4: Illustration of the automaton.

Figure 5: Experimental results of the FixingCopies instances.

Conclusions
- Proposed the first CP formulation for the problem of popular matching involving possibly ties.
- Studied the extended version where additional copies of posts can be added.

References