Popular Matchings in the Weighted Capacitated House Allocation Problem

Colin T.S. Sng* and David F. Manlove†

Department of Computing Science, University of Glasgow, Glasgow G12 8QQ, UK.
Email: {sngts,davidm}@dcs.gla.ac.uk.

Abstract

We consider the problem of finding a popular matching in the *Weighted Capacitated House Allocation problem* (WCHA). An instance of WCHA involves a set of agents and a set of houses. Each agent has a positive weight indicating his priority, and a preference list in which a subset of houses are ranked in strict order. Each house has a capacity that indicates the maximum number of agents who could be matched to it. A matching M of agents to houses is *popular* if there is no other matching M' such that the total weight of the agents who prefer their allocation in M' to that in M exceeds the total weight of the agents who prefer their allocation in M to that in M'. Here, we give an $O(\sqrt{C}n_1 + m)$ algorithm to determine if an instance of WCHA admits a popular matching, and if so, to find a largest such matching, where C is the total capacity of the houses, n_1 is the number of agents, and m is the total length of the agents' preference lists.

1 Introduction

An instance I of the *Weighted Capacitated House Allocation problem*, abbreviated by WCHA, involves a bipartite graph $G = (A, H, E)$, where $A = \{a_1, a_2, ..., a_{n_1}\}$ is the set of *agents*, $H = \{h_1, h_2, ..., h_{n_2}\}$ is the set of *houses* and E is the set of edges in G. We let $n = n_1 + n_2$ and $m = |E|$. Each agent $a \in A$ ranks in strict order a subset of H (the *acceptable* houses for a) represented by his *preference list*, and E consists of all pairs (a, h_j) such that the house h_j appears in the preference list of agent a. We also create a unique *last resort* house $l(a)$ for each a and append $l(a)$ to a's preference list. We also henceforth assume that G contains the vertex $l(a)$ and the edge $(a, l(a))$ for each $a \in A$. Every agent a has a positive weight $w(a)$ indicating the priority of the agent, and we partition A into sets P_1,P_2, ..., P_k, such that the weight of agents in P_z is w_z, and $w_1 > w_2 > ... > w_k > 0$. For each agent $a \in A$, a has *priority* z if $a \in P_z$, and in such a case we let $P(a) = z$. Each house $h_j \in H$ has a *capacity* $c_j \geq 1$ which indicates the maximum number of agents that may be assigned to it. We assume that $m \geq \max\{n_1, n_2\}$, i.e. no agent has an empty preference list and each house is acceptable to at least one agent. We also assume that $c_j \leq n_1$ for each $h_j \in H$. Let $C = \sum_{j=1}^{n_2} c_j$ denote the sum of the capacities of the houses. A *matching* M in I is a subset of E such that (i) each agent is assigned to at most one house in M, and (ii) each house $h_j \in H$ is assigned to at most c_j agents in M. We say that a house $h_j \in H$ is *full* in M if $|M(h_j)| = c_j$ and *undersubscribed* in

*Supported by Department of Computing Science and ORSAS scholarships.

†Supported by an RSE/Scottish Executive Personal Research Fellowship and by EPSRC grant EP/E011993/1.

M if $|M(h_j)| < c_j$. If an agent $a \in A$ is matched in M, we denote by $M(a)$ the house that a is matched to in M. We define $M(h_j)$ to be the set of agents matched to h_j in M (thus $M(h_j)$ could be empty). Given two matchings M and M' in I, we say that an agent a *prefers* M' to M if either (i) a is matched in M' and unmatched in M, or (ii) a is matched in both M' and M and prefers $M'(a)$ to $M(a)$. Let $P(M', M)$ denote the set of agents who prefer M' to M. Then, the *satisfaction* of M' with respect to M is defined as $sat(M', M) = \sum_{a \in P(M',M)} w(a) - \sum_{a \in P(M,M')} w(a)$. We say that M' is *more popular than* M if $sat(M', M) > 0$. Furthermore, a matching M in I is *popular* if there is no other matching in I that is more popular than M.

WCHA is an example of a bipartite matching problem with one-sided preferences [1, 2, 9, 3]. These problems have applications in areas such as campus housing allocation in US universities [1], hence the problem name; in assigning probationary teachers to their first posts in Scotland; and in Amazon's DVD rental service. In many situations where the total capacity of the houses is fewer than the number of agents or where we would like to give some agents a better chance of "doing well", the assignment of weights to agents allow us to build up a spectrum of priority levels for agents in the competition for houses. For instance, Amazon can give priority to those members of its DVD rental service who have paid more for privileged status whenever a certain title is limited in stock. A variety of optimality criteria have been defined for such problems. Gärdenfors [8] first introduced the notion of a popular matching in the context of voting theory. Alternatively, *Pareto optimality* [1, 2] is often regarded by economists as a fundamental property to be satisfied. A matching M is *Pareto optimal* if there is no matching M' such that some agent prefers M' to M, and no agent prefers M to M'. Finally, a matching is *rank maximal* [9] if it assigns the maximum number of agents to their first-choice houses, and subject to this, the maximum number of agents to their second-choice houses, and so on. However, Pareto optimal matchings and rank maximal matchings need not be popular.

Popular matchings were considered by Abraham et al. [3] in the context of the *House Allocation problem* (HA) – the special case of WCHA where each house has capacity 1 and each agent has the same priority. They gave an instance of HA in which no popular matching exists and also noted that popular matchings can have different sizes. They described an $O(n + m)$ algorithm for finding a maximum cardinality popular matching (henceforth a maximum popular matching) if one exists, given an instance of HA. They also described an $O(\sqrt{n}m)$ algorithm for HA when preferences may include ties, i.e. *HA with Ties* (HAT).

Several other recent papers have also focused on popular matchings. Chung [6] considered popular matchings in instances of the Stable Roommate problem (a non-bipartite generalisation of HA) and noted that a stable matching is popular; however, the same need not be true in the presence of ties. Mahdian [10] showed that a popular matching exists with high probability when (i) preference lists are random, and (ii) the number of houses is a small multiplicative factor larger than the number of agents. Abraham and Kavitha [4] considered *voting paths* in relation to popular matchings in a dynamic matching market in which agents and houses can enter and leave the market. Manlove and Sng [11] studied a special case of WCHA in which all agents have the same priority – the *Capacitated House Allocation problem* (CHA). They gave an $O(\sqrt{C}n_1 + m)$ algorithm for finding a (maximum) popular matching, if one exists, when preferences are strict, and an $O((\sqrt{C}+n_1)m)$ algorithm when preferences contain ties. Mestre [12] studied a special case of WCHA in which all houses have unitary capacity – the *Weighted House Allocation problem* (WHA). He gave an $O(n + m)$ algorithm for finding a (maximum) popular matching, if one exists, when preferences are strict, and an $O(\min(k\sqrt{n}, n)m)$ algorithm when preferences contain ties.

```
1.    for each h_j ∈ H loop
2.       for i in 1..k loop
3.          f_{i,j} := 0;
4.    for each a ∈ P_1 loop
5.       f(a) := first-ranked house h_j on a's preference list;
6.       f_{1,j} ++;
7.    for each z in 2..k loop
8.       for each a ∈ P_z loop
9.          q := 1;
10.         h_j := house at position q on a's preference list;
11.         while (Σ_{i=1}^{z-1} f_{i,j} ≥ c_j) loop
12.            q ++;
13.            h_j := house at position q on a's preference list;
14.         f(a) := h_j;
15.         f_{z,j} ++;
```

Figure 1: Algorithm Label-f.

In this paper, we consider popular matchings in an instance I of WCHA (i.e. preference lists are strict), which is a natural generalisation of the one-one WHA model. We give a non-trivial extension of the results from [12] to WCHA. We first develop in Section 2 a characterisation of popular matchings in WCHA. Then, in Sections 3.1 and 3.2, we show how to remove certain edges in the underlying graph G which cannot belong to a popular matching. In Section 3.3, we show how this leads to an $O(\sqrt{C}n_1 + m)$ algorithm for finding a popular matching in I, if one exists. Also, we show in Section 3.3 that any alternative straightforward approach using "cloning" to solve WCHA instances leads to slower algorithms than our direct approach. Finally, we show in Section 3.4 how to modify our algorithm to compute a maximum popular matching if one exists, without altering the time complexity.

2 Characterising a popular matching

For each agent $a \in A$, we introduce the notion of a's *f-house* and a's *s-house* denoting these by $f(a)$ and $s(a)$ respectively. Intuitively, $f(a)$ is the most preferred house h_j on a's preference list to which a could be matched in a popular matching. Therefore, f and s stand for *first possible* and *second possible* house, respectively. We use Algorithm Label-f shown in Figure 1 to define $f(a)$ precisely.

Here, we will define the f-houses for all the agents in phases, with each phase corresponding to a priority level P_z. Intuitively, during the course of the algorithm's execution, $f_{i,j}$ will denote the number of agents with priority i whose f-house is defined and equal to h_j. Initially, $f_{i,j} = 0$ for all i $(1 \leq i \leq k)$ and j $(1 \leq j \leq n_2)$. We then define the f-house for each agent as follows. For every agent $a \in P_1$, we let $f(a)$ be the first-ranked house h_j on a's preference list, and we call such a house an f_1-house. Given $2 \leq z \leq k$, for every agent $a \in P_z$, we let $f(a)$ be the most-preferred house h_j on a's preference list such that $\sum_{i=1}^{z-1} f_{i,j} < c_j$ – we call h_j an f_z-house. Once the algorithm has terminated, we let $f_i(h_j)$ denote the set $\{a \in P_i : f(a) = h_j\}$. Then, $f_{i,j} = |f_i(h_j)|$ (possibly $f_{i,j} = 0$). It is straightforward to verify that Algorithm Label-f runs in $O(m)$ time if we use virtual initialisation (described in [5, p.149]) for the steps in lines 1-3. The following example in Figure 2 gives an illustration of the definition of f-houses. Here, the f-houses of the agents are as follows: $f(a_1) = h_1$, $f(a_2) = h_3$, $f(a_3) = h_3$, $f(a_4) = h_4$, $f(a_5) = h_4$ and $f(a_6) = h_4$.

Now, for each $h_j \in H$, let $f(h_j) = \{a \in A : f(a) = h_j\}$ and $f_j = |f(h_j)|$ (possibly

Agent	Priority	Weight	Pref list				House	Capacity
a_1:	1	7	h_1	h_2	h_3		h_1	1
a_2:	2	4	h_1	h_3	h_4		h_2	2
a_3:	2	4	h_3	h_5			h_3	2
a_4:	3	2	h_3	h_1	h_4	h_5	h_4	2
a_5:	3	2	h_1	h_4	h_5		h_5	1
a_6:	3	2	h_4	h_1	h_2			

Figure 2: Example WCHA instance.

$f_j = 0$), i.e. $f(h_j) = \bigcup_{p=1}^{k} f_p(h_j)$. Clearly each h_j may be an f_z-house for more than one priority level z. For every such h_j, let us define d_j to denote the priority level such that

$$d_j = \begin{cases} \max\{r : 0 \leq r \leq k \wedge f_{r,j} \neq 0\}, \text{if } f_j \leq c_j, \\ \max\{r : 0 \leq r \leq k \wedge \sum_{i=1}^{r} f_{i,j} \leq c_j\}, \text{if } f_j > c_j. \end{cases}$$

Note that for every h_j such that $f_j > c_j$, clearly $\bigcup_{p=d_j+1}^{k} f_p(h_j) \neq \emptyset$. Hence, for every such h_j, we define g_j to be the priority level such that $g_j = \max\{r : d_j < r \leq k \wedge f_{r,j} \neq 0\}$. We refer to Figure 2 for illustration. Here, $d_1 = 1$, $d_3 = 2$ and $d_4 = 2$. Note that d_2 and d_5 are not defined, for h_2 and h_5 are not f-houses for any agent. Also, since $f_4 > c_4$, it follows that $g_4 = 3$; however, h_4 is the only f-house h_j such that $f_j > c_j$.

The following lemmas are vital first steps in characterising popular matchings in WCHA (due to space limitations, see [13] for the proofs of all lemmas and theorems).

Lemma 1. *Let M be a popular matching in any given WCHA instance I. Then, for each $h_j \in H$, $\bigcup_{i=1}^{d_j} f_i(h_j) \subseteq M(h_j)$.*

Lemma 2. *Let M be a popular matching in any given WCHA instance I. Then, for each $h_j \in H$, if $f_j > c_j$, then $M(h_j) \setminus \bigcup_{p=1}^{d_j} f_p(h_j) \subseteq f_{g_j}(h_j)$.*

We now define the concept of an s-house for each agent. If $M(a) \neq f(a)$, then as we shall show, $M(a) = s(a)$. Given $1 \leq z \leq k$, for every agent $a \in P_z$, we define $s(a)$ to be the most preferred house h_j on a's preference list such that $h_j \neq f(a)$ and $\sum_{i=1}^{z} f_{i,j} < c_j$. To illustrate s-houses, let us look at the example in Figure 2 again. We may verify from the definition of s-houses that $s(a_1) = h_2$, $s(a_2) = h_4$, $s(a_3) = h_5$, $s(a_4) = h_5$, $s(a_5) = h_5$ and $s(a_6) = h_2$. Clearly, the set of f_i-houses need not be disjoint from the set of s_j-houses for $i \neq j$. Now, since the process of defining s-houses is analogous to the algorithm for defining f-houses, the time complexity for defining s-houses is also $O(m)$. Note that $s(a)$ may not exist if $f(a) = l(a)$. However, all such agents will be matched to their f-houses in any matching since last resort houses are unique to individual agents. Now, it may be shown that a popular matching M will only match an agent a to either $f(a)$ or $s(a)$ as indicated by the next lemma.

Lemma 3. *Let M be a popular matching in any WCHA instance I. Then, every agent $a \in A$ is matched in M to either $f(a)$ or $s(a)$.*

Lemmas 1-3 give rise to the following result.

Theorem 4. *Let M be a popular matching in any given WCHA instance I.*

1. *For every f-house h_j,*
   - *(a) if $f_j \leq c_j$, then $f(h_j) \subseteq M(h_j)$;*
   - *(b) if $f_j > c_j$, then $|M(h_j)| = c_j$, $\bigcup_{p=1}^{d_j} f_p(h_j) \subseteq M(h_j)$, and $M(h_j) \setminus \bigcup_{p=1}^{d_j} f_p(h_j) \subseteq f_{g_j}(h_j)$.*
2. *Every agent a is matched to either $f(a)$ or $s(a)$.*

3 Algorithm for finding a popular matching

Let us form a subgraph G' of G by letting G' contain only two edges for each agent $a \in A$, that is, one to $f(a)$ and the other to $s(a)$. It follows that all popular matchings must be contained in G' by Conditions 1 and 2 of Theorem 4. However, Theorem 4 only gives us necessary conditions for a matching to be popular in an instance of WCHA, since not all matchings in G' satisfying these conditions are popular. For let us consider the example WCHA instance in Figure 2. We have two matchings which satisfy Conditions 1 and 2 of Theorem 4: $M_1 = \{(a_1, h_1), (a_2, h_3), (a_3, h_3), (a_4, h_5), (a_5, h_4), (a_6, h_4)\}$ and $M_2 = \{(a_1, h_1), (a_2, h_3), (a_3, h_3), (a_4, h_4), (a_5, h_5), (a_6, h_4)\}$. However, while M_1 is a popular matching, M_2 is not popular because there exists another matching $M_3 = \{(a_2, h_1), (a_3, h_3), (a_4, h_3), (a_5, h_4), (a_6, h_4)\}$ which gives an improvement in satisfaction of $w(a_2) + w(a_4) + w(a_5) - w(a_1) = 4 + 2 + 2 - 7 > 0$ over M_2. Hence, we will "enforce" the sufficiency of the conditions by removing certain edges in G' that cannot form part of any popular matching in I. We show how to do this by first introducing the notion of a *potential improvement path* or *PIP* in short, which generalises the concept of a *promotion path* from [12] to WCHA.

3.1 Potential improvement paths

Let us now define a matching M that satisfies Conditions 1 and 2 of Theorem 4 to be *well-formed*. Then, a PIP leading out of some f-house h_{i_0} with respect to M is an alternating path $\Pi = \langle h_{i_0}, a_{i_0}, h_{i_1}, a_{i_1}, ..., h_{i_x}, a_{i_x} \rangle$ such that $h_{i_j} = f(a_{i_j})$ and $(a_{i_j}, h_{i_j}) \in M$ for $0 \leq j \leq x$, and a_{i_j} strictly prefers $h_{i_{j+1}}$ to h_{i_j} for $j < x$. A PIP leading out of h_{i_0} always exists, which can be seen as follows. Since h_{i_0} is an f-house and $c_{i_0} \geq 1$, there exists some agent $a_{i_0} \in f(h_{i_0}) \cap M(h_{i_0})$ by Theorem 4. Then, by definition, $\langle h_{i_0}, a_{i_0} \rangle$ is a PIP leading out of h_{i_0}. The next lemma shows that any PIP leading out of h_{i_0} must contain a sequence of agents with strictly decreasing priorities. Hence, the sequence of agents in Π must be distinct since the priority of agents is strictly decreasing.

Lemma 5. *Let M be a well-formed matching. Let $\Pi = \langle h_{i_0}, a_{i_0}, ..., h_{i_x}, a_{i_x} \rangle$ be a potential improvement path with respect to M leading out of h_{i_0} as defined above. Then, $P(a_{i_{j+1}}) < P(a_{i_j})$ for $0 \leq j < x$.*

Let us define the cost of Π to be $cost(\Pi) = w(a_{i_x}) - w(a_{i_{x-1}}) - ... - w(a_{i_0})$. Note that $cost(\Pi) = w(a_{i_0})$ if $x = 0$. We now motivate the notion of a PIP as follows. Let us suppose that there exists some agent a_r who prefers h_{i_0} to $M(a_r)$. The next lemma shows that any such agent cannot belong to Π. Now, if $cost(\Pi) < w(a_r)$, we can conclude that the well-formed matching M is not popular because we can promote a_r to h_j, and use the PIP to promote each a_{i_j} to $h_{i_{j+1}}$ for all $j < x$ and demote a_{i_x} to $l(a_{i_x})$ to obtain a new matching that is more popular than M.

Lemma 6. *Let M be a well-formed matching. Let $\Pi = \langle h_{i_0}, a_{i_0}, ..., h_{i_x}, a_{i_x} \rangle$ be a potential improvement path with respect to M leading out of h_{i_0} as defined above. Then, any agent a who prefers h_{i_0} to $M(a)$ does not belong to Π.*

3.2 Pruning the graph

Let us now introduce Algorithm Prune-WCHA, as given in Figure 3, which will enable us to remove certain edges in G' that cannot be part of any popular matching. The algorithm is divided into two stages, and the first stage is carried out in phases, with each phase corresponding to a priority level P_z. Intuitively, in each phase in the first stage, we compute the costs of PIPs and determine the minimum of these for each f-house h_j, and then use

these values to identify and remove certain edges incident to f-houses in G' that cannot belong to any popular matching. Based on the minimum values of PIPs calculated for f-houses in the first stage, we then identify and remove in the second stage edges incident to s-houses in G' that cannot belong to any popular matching. Let G'' denote the graph obtained from G' once the algorithm terminates (following these pruning operations). The removal of these edges will ensure that any well-formed matching in G'' is popular. Over the phases of execution, certain conditions may arise which signal to the algorithm that no popular matching exists.

Recall that h_j may be an f-house for more than one priority level, and h_j may be an f-house for more than one agent for each priority level. In the algorithm, we will use $\lambda_z(h_j)$ to compute the minimum cost of a PIP leading out of h_j taken over all well-formed matchings in G'' such that (a_r, h_j) is the first edge for some $a_r \in P_z$. We will also use $\lambda(h_j)$ to compute the minimum cost taken over all $\lambda_z(h_j)$. Let $\Pi_{min}(h_j)$ denote a PIP with minimum cost leading out of h_j taken over all well-formed matchings in G''. Let $cost(\Pi_{min}(h_j))$ denote the cost of this path. Then, the final value of $\lambda(h_j)$ in the execution of the algorithm gives us the value of $cost(\Pi_{min}(h_j))$. For any agent $a_s \in A$, let S contain the set of houses on a_s's preference list that a_s prefers to $f(a_s)$. If $S \neq \emptyset$, we will use $\lambda_{min}(a_s, f(a_s))$ within the algorithm to compute the minimum cost of a PIP out of h_q taken over all $h_q \in S$, and over all well-formed matchings in G''; otherwise, $\lambda_{min}(a_s, f(a_s)) = \infty$. Similarly, let R contain the set of houses on a_s's preference list after $f(a_s)$ that a_s prefers to $s(a_s)$. If $R \neq \emptyset$, we will use $\lambda_{min}(a_s, s(a_s))$ within the algorithm to compute the minimum cost of a PIP out of h_q taken over all $h_q \in R$, and over all well-formed matchings in G''; otherwise, $\lambda_{min}(a_s, s(a_s)) = \infty$.

The next three lemmas are vital results which establish the correctness of the algorithm.

Lemma 7. *Any edge removed by Algorithm Prune-WCHA cannot belong to a popular matching.*

Lemma 8. *If Algorithm Prune-WCHA reports that no popular matching exists, then there does not exist any well-formed matching in G' that is popular.*

Lemma 9. *Suppose that Algorithm Prune-WCHA does not state that no popular matching exists. Let M be a well-formed matching in the pruned graph G''. Then, M is popular.*

We now use the example in Figure 2 to illustrate our algorithm. After the first stage, we have $\lambda(h_1) = 7$, $\lambda(h_3) = 3$ and $\lambda(h_4) = 2$. We remove the edges (a_1, h_2) in phase 1, and (a_2, h_4) and (a_3, h_5) in phase 2 of the first stage (in line 17) since a_1 belongs to $f_{d_1}(h_1)$, and a_2 and a_3 belong to $f_{d_3}(h_3)$ respectively. We also remove the edge (a_4, h_4) in phase 3 of the first stage (in lines 20-21) since $\lambda_{min}(a_4, h_4) = 3 < 2w(a_4)$. No further edges are removed in the second stage.

3.3 Finding a popular matching

We are now left with the task of finding a well-formed matching M in G'' in order to find a popular matching if one exists. Note that the removal of edges from G' by Algorithm Prune-WCHA effectively reduces the problem to that of finding a popular matching in an instance of CHA. For let us consider the problem of trying to match each f-house h_j so that h_j satisfies the conditions of a well-formed matching. Now, if $f_j \leq c_j$, then ensuring that $\bigcup_{p=1}^{d_j} f_p(h_j) \subseteq M(h_j)$ is equivalent to ensuring Condition 1(a) of Theorem 1 of [11]. On the other hand, if $f_j > c_j$, we need to ensure that those agents with the correct priorities are matched to h_j in M, i.e. there does not exist any agent $a \in \bigcup_{p=1}^{d_j} f_p(h_j) \backslash M(h_j)$. Now, since line 17 in the first stage of Algorithm Prune-WCHA ensures the removal of the edge

```
1.   for each f-house h loop
2.       λ(h) := w_1; /* a suitable upper bound */
3.   - - - - - first stage - - - - - - -
4.   for z in 1..k loop
5.       for each a ∈ P_z loop
6.           S := {h ∈ H : a prefers h to f(a)};
7.           if S ≠ ∅ then
8.               λ_min(a, f(a)) := min {λ(h) : h ∈ S};
9.           else
10.              λ_min(a, f(a)) := ∞; /* a suitable default value */
11.          if λ_min(a, f(a)) < w_z then
12.              return "No popular matching exists";
13.      for each f_z-house h_j loop
14.          f'_z(h_j) := f_z(h_j);
15.          if z ≤ d_j
16.              for each a ∈ f'_z(h_j)
17.                  remove (a, s(a)) from G';
18.          else (// z > d_j )
19.              for each a ∈ f'_z(h_j) such that λ_min(a, h_j) < 2w_z loop
20.                  remove (a, h_j) from G';
21.                  remove a from f'_z(h_j);
22.              if f'_z(h_j) = ∅ then
23.                  return "No popular matching exists.";
24.          λ_z(h_j) := min(w_z, min {λ_min(a, h_j) − w_z : a ∈ f'_z(h_j)});
25.          λ(h_j) := min(λ(h_j), λ_z(h_j));
26.          if z = g_j and λ(h_j) < w_z then
27.              return "No popular matching exists.";
28.  - - - - - - second stage - - - - - - - -
29.  for each a ∈ A loop
30.      h_l := s(a);
31.      R := {h ∈ H : a prefers h to h_l} \(S ∪ {f(a)});
32.      if R ≠ ∅ then
33.          λ_min(a, h_l) := min {λ(h) : h ∈ R};
34.      else
35.          λ_min(a, h_l) := ∞;/* a suitable default value */
36.      if λ_min(a, h_l) < w(a) or f_l ≥ c_l then
37.          remove (a, h_l) from G';
```

Figure 3: Algorithm Prune-WCHA for removing edges from G'

$(a, s(a))$ of every such a where $a \in \bigcup_{p=1}^{d_j} f_p(h_j)$, a must be matched to $f(a)$ if an *agent-complete* matching (i.e. a matching in which all agents are matched) is to exist. This is equivalent to the work done by lines 10-18 of Algorithm Popular-CHA of [11], which tries to find an agent-complete matching and reports that no popular matching exists if unsuccessful. Lastly, we need to ensure that each agent is matched to either $f(a)$ or $s(a)$ and it is evident that Algorithm Popular-CHA also does this. Hence, we can find a popular matching in WCHA, if one exists, by running Algorithm Popular-CHA on the pruned graph G''. As illustration, if we run Algorithm Popular-CHA on the example in Figure 2 after edge removals through Algorithm Prune-WCHA, then Algorithm Popular-CHA will return the following matching $M = \{(a_1, h_1), (a_2, h_3), (a_3, h_3), (a_4, h_5), (a_5, h_4), (a_6, h_4)\}$ which may be verified to be popular.

Let us now consider the time taken to find a popular matching in an instance of WCHA, or to report that no such matching exists. First of all, it takes $O(m)$ time to define the f- and s-houses. Algorithm Prune-WCHA may be verified to run in $O(m)$ time overall (see [13]). Now, it takes $O(\sqrt{C}n_1 + m)$ time, using Algorithm Popular-CHA, to find a

well-formed matching (if one exists) in G'', where C is the total capacity of the houses. This gives us the following.

Theorem 10. *Let I be an instance of WCHA. Then, we can find a popular matching in I, or determine that none exists, in $O(\sqrt{C}n_1 + m)$ time.*

"Cloning" versus our direct approach

A straightforward solution to finding a popular matching, given an instance I of WCHA, may be to use "cloning" to create an instance J of WHA where preference lists are allowed to contain ties, and then to apply the $O(\min(k\sqrt{n}, n)m)$ algorithm of [12] to J. Firstly, we create c_j clones $h_j^1, h_j^2, ..., h_j^{c_j}$ of each house h_j in I, where each clone has a capacity of 1. In addition, we replace each occurrence of h_j in a given agent's preference list with the sequence $h_j^1, h_j^2, ..., h_j^{c_j}$, the elements of which are listed in a single tie at the point where h_j appears. Let G_J denote the underlying graph of J. Then, G_J contains $n' = n_1 + C$ nodes. For each $a_i \in A$, let A_i denote the set of acceptable houses for a_i, and let $c_{min} = \min\{c_j : h_j \in H\}$. Then the number of edges in G_J is $m' = \sum_{a_i \in A} \sum_{h_j \in A_i} c_j \geq mc_{min}$. Hence, the complexity of applying the algorithm of [12] to J is $\Omega(\min(k\sqrt{n_1 + C}, n_1 + C)mc_{min})$. Now, the complexity of our algorithm may be rewritten as $O(\sqrt{C}n_1)$ or $O(m)$ depending on which component dominates the running time. If $n_1 + C \geq k\sqrt{n_1 + C}$, then the cloning approach takes $\Omega(k\sqrt{n_1 + C}mc_{min}))$ time which is slower than our algorithm by a factor of $\Omega(kc_{min})$. Otherwise, if $n_1 + C < k\sqrt{n_1 + C}$, then the cloning approach takes $\Omega(mc_{min}(n_1 + C))$ time which is slower than our algorithm by a factor of $\Omega(\sqrt{C}c_{min})$. It follows that the cloning method is slower than our direct approach for all possible cases.

3.4 Finding a maximum popular matching

It remains to consider the problem of finding a maximum popular matching in WCHA. Let us run Algorithm Label-f and Algorithm Prune-WCHA as before to define f- and s-houses and to delete certain edges which cannot belong to any popular matching. We then adopt a similar algorithm to that in [11] as follows. That is, let A_1 be the set of all agents a with $s(a) = l(a)$, and let $A_2 = A \backslash A_1$. Our objective is to find a well-formed matching in G'' which minimises the number of A_1-agents who are matched to their last resort house. We let A' denote the set $\left\{a \in \bigcup_{p=1}^{d_j} f_p(h_j)\right\}$. We begin by carrying out a pre-processing step on G'' to compute a matching M_0 that matches each agent in A' to his f-house. We then try to find a maximum matching M' in G'' that only involves the $A_2 \backslash A'$ agents that remain unmatched after pre-processing and their incident edges. If M' is not an agent-complete matching of $A_2 \backslash A'$ agents that remain unmatched after pre-processing, then clearly I admits no popular matching. Otherwise, we remove all edges in G'' that are incident to a last resort house, and try to match additional A_1-agents to their f-houses by repeatedly finding an augmenting path with respect to M' using Gabow's algorithm [7] in a similar approach to that for CHA [11]. Let M'' be the matching obtained by augmenting M'. If any A_1-agent remains unmatched at the end of this step, we simply assign him to his last resort house, to obtain an agent-complete matching of $A \backslash A'$ agents in G''. Let $M = M_0 \cup M''$. If any agent a belonging to $A \backslash A'$ is not matched to his f-house h_j but h_j is undersubscribed in M, we promote a from $M(a)$ to h_j. Then, clearly M will be a well-formed matching in G'', and hence popular by Lemma 9. It follows that M is a maximum popular matching, giving the following theorem.

Theorem 11. *Given an instance of WCHA, we can find a maximum popular matching, or determine that none exists, in $O(\sqrt{C}n_1 + m)$ time.*

Acknowledgement

We would like to thank Rob Irving for helpful discussions concerning this paper.

References

[1] A. Abdulkadiroğlu and T. Sönmez. Random serial dictatorship and the core from ra ndom endowments in house allocation problems. *Econometrica*, 66:689–701, 1998.

[2] D.J. Abraham, K. Cechlárová, D.F. Manlove, and K. Mehlhorn. Pareto optimality in house allocation problems. In *Proceedings of ISAAC '04*, vol. 3341 of *Lecture Notes in Computer Science*, pp. 3–15. Springer, 2004.

[3] D.J. Abraham, R.W. Irving, T. Kavitha, and K. Mehlhorn. Popular matchings. In *Proceedings of SODA 2005*, pp. 424–432. ACM-SIAM, 2005.

[4] D.J. Abraham, T. Kavitha. Dynamic matching markets and voting paths. In *Proceedings of SWAT 2006*, vol. 4059 of *Lecture Notes in Computer Science*, pp. 65–76. Springer, 2006.

[5] G. Brassard, P. Bratley. Fundamentals of Algorithmics. Prentice-Hall, 1996.

[6] K.S. Chung. On the Existence of Stable Roommate Matchings. In *Games and Economic Behavior*, 33:206–230, 2000.

[7] H.N. Gabow. An efficient reduction techique for degree-constrained subgraph and bidirected network flow problems. In *Proceedings of STOC 1983*, pp. 448–456. ACM, 1983.

[8] P. Gärdenfors. Match making: assignments based on bilateral preferences. vol. 20, pp. 166–173. Behavioural Science, 1975.

[9] R.W. Irving, T. Kavitha, K. Mehlhorn, D. Michail and K. Paluch. Rank maximal matchings. In *Proceedings of SODA 2004*, pp. 68–75. ACM-SIAM, 2004.

[10] M. Mahdian. Random popular matchings. In *Proceedings of EC 2006*, pp. 238–242. ACM, 2006.

[11] D.F. Manlove, C.T.S. Sng. Popular matchings in the capacitated house allocation problem. In *Proceedings of ESA 2006*, vol. 4168 of *Lecture Notes in Computer Science*, pp. 492–503. Springer-Verlag, 2006.

[12] J. Mestre. Weighted poopular matchings. In *Proceedings of ICALP 2006*, vol. 4051 of *Lecture Notes in Computer Science*, pp. 715–726. Springer-Verlag, 2006.

[13] C.T.S. Sng, D.F. Manlove. Popular matchings in the weighted capacitated house allocation problem. Technical Report TR-2007-254, University of Glasgow, Department of Computing Science, July 2007.