Sng, C.T.S. and Manlove, D.F. (2010) *Popular matchings in the weighted capacitated house allocation problem.* Journal of Discrete Algorithms, 8 (2). pp. 102-116. ISSN 1570-8667

# 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* (WCHA) involves a set of *agents* $A = \{a_1, a_2, ..., a_{n_1}\}$ and a set of *houses* $H = \{h_1, h_2, ..., h_{n_2}\}$. Each agent $a \in A$ ranks in strict order a subset of $H$ (the *acceptable* houses for $a$) represented by his *preference list*. We also create a unique *last resort* house $l(a)$ for each $a$ and append $l(a)$ to $a$'s preference list. Every agent $a$ also has a positive weight $w(a)$ indicating $a$'s *priority*, 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 $a \in A$, we say that $a$ has *priority* $z$ if $a \in P_z$, and we use $P(a)$ to denote the priority of $a$. 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.

The *underlying graph* of $I$ is the bipartite graph $G = (A, H \cup L, E)$, where $L$ is the set of last resort houses, and $E$ comprises all pairs $(a, h_j)$ such that house $h_j$ appears in the preference list of agent $a$ (note that this includes the pairs $(a, l(a))$ for each agent $a$.) We let $n = n_1 + n_2$ and $m = |E|$. We assume that $m \geq \max\{n_1, n_2 + L\}$, 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 $M$.

*A preliminary version of this paper appeared in *Proceedings of ACiD 2007* [18].
†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.

We say that a house $h_j \in H$ is *full* in $M$ if $|M(h_j)| = c_j$, and *undersubscribed* in $M$ 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 $sat(M', M) > 0$. A matching $M$ in $I$ is *popular* if there is no other matching in $I$ that is more popular than $M$.

**Motivation**

WCHA is an example of a bipartite matching problem with one-sided preferences [1, 2, 10, 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. The assignment of weights to agents allows us to build up a spectrum of priority levels for them in the competition for houses in situations where the total capacity of the houses is less than the number of agents. In turn, this gives some agents a better chance of "doing well". For instance, the assignment of weights can enable DVD rental companies like Amazon to give priority to those members who have paid more for privileged status whenever a certain title is limited in stock. Alternatively, weights may be assigned to candidates in job markets based on objective criteria such as academic results or relevant work experience.

A variety of optimality criteria have been defined for bipartite matching problems with one-sided preferences. Gärdenfors [8] first introduced the notion of a popular matching (referring to this concept as a *majority assignment*) in the context of the Stable Marriage problem. We remark that the *more popular than* concept can be traced back even further to the Condorcet voting protocol. 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 $M'$. Clearly a popular matching is Pareto optimal. Finally, a matching is *rank-maximal* [10] 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.

**Previous work**

Popular matchings were first considered from an algorithmic point of view 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. The authors also formulated 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 the case where preferences may include ties, i.e., *HA with Ties* (HAT).

Several other papers have also focused on popular matchings. Chung [6] considered popular matchings in instances of the Stable Roommates 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. In the HA context, Mahdian [13] 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. In the context of HAT, 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 [14] studied the *Capacitated House Allocation problem* (CHA) – this is the special case of WCHA in which all agents have the same priority. 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 [17] studied the *Weighted House Allocation problem* (WHA) – this is the special case of WCHA in which all houses have unitary capacity. 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. Kavitha and Shah [12] gave an $O(n^\omega)$ randomized algorithm for finding a popular matching or reporting that none exists, given an instance of HAT, where $\omega < 2.376$ is the exponent of matrix multiplication.

To cope with the possible non-existence of a popular matching, McCutchen [15] defined two notions of a matching that is, in some sense, "as popular as possible", namely a *least-unpopularity-factor matching* and a *least-unpopularity-margin matching*, for instances of HA and HAT. McCutchen proved that computing either type of matching is NP-hard, even if preference lists are strictly ordered. Huang et al. [9] gave an $O(\sqrt{n}m)$ algorithm for finding a matching $M$ with unpopularity factor 2, provided that a certain graph admits a matching in which all agents are matched. They also generalised this result by describing a sequence of graphs $H_2, H_3, \ldots, H_r$ such that if $H_r$ admits a matching in which all agents are matched, a matching $M$ can be computed in $O(r\sqrt{n}m)$ time with unpopularity factor at most $r - 1$ and unpopularity margin at most $n(1 - \frac{2}{r})$.

Recently, Kavitha and Nasre [11] gave an $O(n^2 + m)$ algorithm for the problem of computing an optimal popular matching (assuming a popular matching exists), given an instance of HA, where "optimal" includes the rank-maximal, *fair* (i.e., minimise the number of agents who obtain their $r$th choice house, and subject to this, minimise the number who obtain their $(r-1)$th choice, and so on, where $r$ is the maximum length of an agent's preference list) and *minimum cost* (i.e., minimise the sum of the ranks of the agents' assigned houses in their preference lists) criteria. McDermid and Irving [16] gave a characterisation of the set of popular matchings for an HA instance in terms of the so-called *switching graph*, which is computable in linear time from the preference lists. They showed that this structure can be exploited to yield efficient algorithms for a range of associated problems, including the counting and enumeration of the set of popular matchings, generation of a popular matching uniformly at random, finding all (agent, house) pairs that can occur in a popular matching, and computing popular matchings that satisfy various additional optimality criteria, including the rank-maximal, fair and minimum cost criteria described above. The algorithm of McDermid and Irving [16] for computing a minimum cost popular matching runs in $O(n + m)$ time, whilst their algorithms for computing a rank-maximal or a fair popular matching have $O(n \log n + m)$ complexity; each of these running times improves on that of Kavitha and Nasre [11] for the same problems.

**Our results**

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 [17] to WCHA. We first develop in Section 2 necessary conditions for a matching to be popular in a WCHA instance $I$. Then, in Sections 3.1 and 3.2, we define a structure in the underlying graph of $I$ that enables us to identify certain edges that cannot belong to a popular matching, giving correctness proofs

**Algorithm 1** Algorithm Label-f

```
for each h_j ∈ H do
    for i in 1..k do
        f_{i,j} := 0;
    end for
end for
for each a ∈ P_1 do
    f(a) := first-ranked house h_j on a's preference list;
    f_{1,j} ++;
end for
for z in 2..k do
    for each a ∈ P_z do
        q := 1;
        h_j := house at position q on a's preference list;
        while (\sum_{p=1}^{z-1} f_{p,j} ≥ c_j) do
            q ++;
            h_j := house at position q on a's preference list;
        end while
        f(a) := h_j;
        f_{z,j} ++;
    end for
end for
```

in Section 3.3. We then use these two results in conjunction to construct in Section 3.4 an $O(\sqrt{C}n_1 + m)$ time algorithm for finding a popular matching in $I$ or reporting that none exists. In Section 3.5 we show how to modify this algorithm to compute a maximum popular matching if one exists, without altering the time complexity. We remark that a straightforward solution to the problem of finding a maximum popular matching in $I$ may be to use "cloning". Informally, this entails creating $c_j$ clones for each house $h_j$ to obtain an instance $J$ of WHAT (WHA with ties), and then applying the algorithm of [17] for WHAT to $J$. However, we will show in Section 3.6 that this approach leads to a slower algorithm than our direct approach.

## 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. Agent $a$ prefers $f(a)$ to $s(a)$, and as we will show, these are the only two houses to which $a$ could be matched in a popular matching. We use Algorithm Label-f shown in Algorithm 1 to define $f(a)$ precisely (the definition of $s(a)$ will follow later in this section.)

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 $i$ $(1 \le i \le k)$ and $j$ $(1 \le j \le n_2)$. 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 $2 \le z \le k$, 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 $\sum_{p=1}^{z-1} f_{p,j} < c_j$ – we call $h_j$ an $f_z$-*house*. Clearly, the algorithm must terminate due to the presence of a unique last resort house at the end of each agent's preference list. 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 $f_{i,j} = 0$). Here, and henceforth throughout this paper, any reference to $f_{i,j}$ refers to the value of this variable upon termination of Algorithm Label-f.

| 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 1: An instance $I_1$ of WCHA

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 1-5. The example in Figure 1 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 $f_j = 0$), i.e., $f(h_j) = \bigcup_{p=1}^{k} f_p(h_j)$. Define $h_j \in H$ to be an *f-house* if $f_j > 0$. Clearly each $h_j$ may be an $f_z$-house for more than one priority level $z$. For each $h_j \in H$ such that $f_j > 0$, let $d_j$ be a priority level defined as follows:

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

Intuitively, if $f_j \leq c_j$ then $d_j$ is the maximum priority level $r$ such that $f(a) = h_j$ for some $a \in P_r$. If $f_j > c_j$ then $d_j$ is the maximum priority level $r$ such that the total number of agents $a$ satisfying $f(a) = h_j$ and $P(a) \leq r$ is less than $c_j$. Note that for every $h_j$ such that $f_j > c_j$, clearly $\sum_{i=1}^{d_j+1} f_{i,j} \geq c_j$, so $f_{d_j+1,j} > 0$. However it is impossible that $f_{i,j} > 0$ for some $i > d_j + 1$ by definition of an $f$-house. It follows that $f_{d_j+1,j} > c_j - \sum_{i=1}^{d_j} f_{i,j}$.

We refer to Figure 1 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 $f_{3,4} > c_4 - (f_{1,4} + f_{2,4})$.

We now work towards obtaining a characterisation of popular matchings in WCHA. We begin with the following technical lemma.

**Lemma 1.** *Let $M$ be a matching in a WCHA instance $I$. Let $h_j \in H$ be an f-house, and let $1 \leq i \leq d_j$. Suppose $h_j$ is full in $M$ and $\bigcup_{p=1}^{i-1} f_p(h_j) \subseteq M(h_j)$ but $f_i(h_j) \nsubseteq M(h_j)$. Then $M(h_j) \backslash \bigcup_{p=1}^{i} f_p(h_j) \neq \emptyset$.*

*Proof.* Let $F = \bigcup_{p=1}^{i} f_p(h_j)$. Clearly, $|F| \leq \sum_{p=1}^{d_j} f_{p,j} \leq c_j$. Since $|F \backslash M(h_j)| > 0$, it follows that $|M(h_j) \backslash F| = |M(h_j)| - |F| + |F \backslash M(h_j)| > 0$. □

The next three lemmas contribute to the characterisation of popular matchings in WCHA.

**Lemma 2.** *Let $M$ be a popular matching in any given WCHA instance $I$ and let $a \in A$ be any agent. Then, $a$ cannot be assigned to a house better than $f(a)$ in $M$.*

*Proof.* Let $a$ be an agent whose priority index is lowest (i.e., $a$ has greatest weight and highest priority) such that $a$ is assigned to house $h_j$ in $M$ and $a$ prefers $h_j$ to $f(a) = h_l$. Let $a \in P_i$ so that $\sum_{p=1}^{i-1} f_{p,j} \geq c_j$ by definition of $f(a)$ as $a$'s $f$-house. Clearly, there must be no agent $a'$ such that $a' \in P_z$ where $z \geq i$ and $f(a') = h_j$, for otherwise $\sum_{p=1}^{z-1} f_{p,j} < c_j$, a contradiction. Let $a'$ be any agent with priority level $z < i$ such that $a' \in f(h_j) \backslash M(h_j)$ – there must exist such an agent since $\bigcup_{p=i}^{k} f_p(h_j) = \emptyset$ and $\sum_{p=1}^{i-1} f_{p,j} \geq c_j$ and $a \in M(h_j)$.

Then, by choice of $a$, $a'$ is assigned in $M$ to a house worse than $f(a')$. However, this means that we can promote $a'$ to $f(a')$ and demote $a$ to $l(a)$ to obtain a matching whose improvement in satisfaction is $w_z - w_i > 0$, a contradiction. □

**Lemma 3.** *Let $M$ be a popular matching in any given WCHA instance $I$. Then, for each f-house $h_j \in H$, $\bigcup_{i=1}^{d_j} f_i(h_j) \subseteq M(h_j)$.*

*Proof.* Given $1 \leq i \leq d_j$, we will prove by induction on $i$ that $f_i(h_j) \subseteq M(h_j)$.

For the base case, let $i = 1$. Suppose that $f_1(h_j) \not\subseteq M(h_j)$. Then, there exists some agent $a_r \in f_1(h_j) \backslash M(h_j)$. By definition of an $f_1$-house, $h_j$ must be the first house on $a_r$'s preference list. Hence, $a_r$ prefers to be assigned to $h_j$ than $M(a_r)$. Clearly, if $h_j$ is undersubscribed in $M$, we can promote $a_r$ to $h_j$ to obtain a matching more popular than $M$, a contradiction. Hence, $h_j$ is full in $M$. Choose any $a_s \in M(h_j) \backslash f_1(h_j)$ (which must exist by Lemma 1). Since $a_s \notin f_1(h_j)$, either (i) $a_s$ has priority $> 1$, or (ii) $a_s$ has priority 1 but $f(a_s) = h_l \neq h_j$. In subcase (i), we can promote $a_r$ to $h_j$ and demote $a_s$ to $l(a_s)$ to obtain a more popular matching. In subcase (ii), since $f(a_s) = h_l$, it follows by Lemma 2 that $a_s$ prefers to be assigned to $h_l$ than $h_j$. Now, if $h_l$ is undersubscribed in $M$, we can promote $a_r$ to $h_j$ and promote $a_s$ to $h_l$ to obtain a more popular matching. Hence, $h_l$ is full in $M$. If $h_l = M(a_r)$, then we can then promote $a_r$ to $h_j$ and promote $a_s$ to $h_l$ to obtain a more popular matching. Otherwise, choose any $a_t \in M(h_l)$. Clearly, $a_t \neq a_r$. We can then promote $a_r$ to $h_j$, promote $a_s$ to $h_l$, and demote $a_t$ to $l(a_t)$ to obtain a matching whose improvement in satisfaction is $w_1 + w_1 - w(a_t) > 0$.

For the inductive case, assume that $2 \leq i \leq d_j$, and if $q < i$, then $f_q(h_j) \subseteq M(h_j)$ for all $h_j \in H$. Suppose for a contradiction that $f_i(h_j) \not\subseteq M(h_j)$. Then, there exists some $a_r \in f_i(h_j) \backslash M(h_j)$. Now, since $f(a_r) = h_j$, it follows by Lemma 2 that $a_r$ must prefer to be assigned to $h_j$ than $M(a_r)$. Thus, if $h_j$ is undersubscribed in $M$, we can promote $a_r$ to $h_j$ to obtain a more popular matching than $M$, a contradiction. Hence, $h_j$ is full in $M$. Choose any $a_s \in M(h_j) \backslash \bigcup_{p=1}^{i} f_p(h_j)$ which must exist by Lemma 1. Since $a_s \notin \bigcup_{p=1}^{i} f_p(h_j)$, either (i) $a_s$ has priority $> i$, or (ii) $a_s$ has priority $\leq i$ but $f(a_s) = h_l \neq h_j$.

In subcase (i), we can promote $a_r$ to $h_j$ and demote $a_s$ to $l(a_s)$ to obtain a more popular matching than $M$, a contradiction. In subcase (ii), suppose that $a_s$ has priority $z < i$. Then $h_l$ is an $f_z$-house so that $a_s \in f_z(h_l)$. However, this is a contradiction since by the inductive hypothesis $f_z(h_l) \subseteq M(h_l)$, but $M(a_s) \neq h_l$. Thus, $a_s$ has priority $i$ and $a_s \in f_i(h_l)$. Clearly, since $f(a_s) = h_l$, it follows by Lemma 2 that $a_s$ must prefer to be assigned to $h_l$ than $h_j$. Thus, if $h_l$ is undersubscribed, we can promote $a_r$ to $h_j$ and promote $a_s$ to $h_l$ to obtain a more popular matching than $M$, a contradiction. Hence $h_l$ is full. If $h_l = M(a_r)$, then we can promote $a_r$ to $h_j$ and promote $a_s$ to $h_l$ to obtain a more popular matching. Otherwise, $h_l \neq M(a_r)$. We will show how to choose $a_t \in M(h_l)$. Since $f(a_s) = h_l$ and $2 \leq i \leq k$, by our definition of $f$-houses, $h_l$ must be the most preferred house on $a_s$'s preference list such that $\sum_{p=1}^{i-1} f_{p,l} < c_l$.

Now, by the inductive hypothesis, it must be the case that $\bigcup_{p=1}^{i-1} f_p(h_l) \subseteq M(h_l)$. Since $\sum_{p=1}^{i-1} f_{p,l} < c_l$ and $h_l$ is full, it follows that $\bigcup_{p=1}^{i-1} f_p(h_l) \subset M(h_l)$. Hence, it must be the case that $M(h_l) \backslash \bigcup_{p=1}^{i-1} f_p(h_l) \neq \emptyset$. It follows that there exists some agent $a_t \in M(h_l) \backslash \bigcup_{p=1}^{i-1} f_p(h_l)$ and, either (i) $a_t \in \bigcup_{p=i}^{k} f_p(h_l)$ or (ii) $a_t \notin f(h_l)$. Clearly, in case (ii), $a_t$ has priority $\geq i$ by a similar argument for $a_s$. For, if $a_t$ has priority $z < i$, then by the inductive hypothesis, since $h_{l'} = f(a_t)$ is an $f_z$-house and $a_t \in f_z(h_{l'})$, it follows that $f_z(h_{l'}) \subseteq M(h_{l'})$. However, this gives a contradiction since $M(a_t) \neq h_{l'}$. Hence, $a_t$ has priority $\geq i$ in both cases (i) and (ii). We can then promote $a_r$ to $h_j$, promote $a_s$ to $h_l$ and demote $a_t$ to $l(a_t)$ to obtain a matching whose improvement in satisfaction is

$w_i + w_i - w(a_t) > 0$, a contradiction. □

**Lemma 4.** *Let $M$ be a popular matching in any given WCHA instance $I$. Then, for each f-house $h_j \in H$, if $f_j > c_j$, then $M(h_j) \backslash \bigcup_{p=1}^{d_j} f_p(h_j) \subseteq f_{d_j+1}(h_j)$.*

*Proof.* Clearly, $f_{d_j+1,j} > c_j - \sum_{p=1}^{d_j} f_{p,j}$. It follows by Lemma 3 that $\bigcup_{p=1}^{d_j} f_p(h_j) \subseteq M(h_j)$ so that no matter whether $h_j$ is full or undersubscribed, $f_{d_j+1}(h_j) \nsubseteq M(h_j)$. Hence, there exists some agent $a_r$ such that $a_r \in f_{d_j+1}(h_j) \backslash M(h_j)$. Note that $a_r$ has priority $d_j + 1$. Clearly, since $f(a_r) = h_j$, $a_r$ must prefer to be assigned to $h_j$ rather than $M(a_r)$ by Lemma 2. Hence, if $h_j$ is undersubscribed, we can promote $a_r$ to $h_j$ to obtain a more popular matching than $M$, a contradiction. It follows that $h_j$ is full. We will show that $M(h_j) \backslash \bigcup_{p=1}^{d_j} f_p(h_j) \subseteq f_{d_j+1}(h_j)$.

If $d_j = 0$, then it must be the case that $f_{1,j} > c_j$ and $a_r \in f_1(h_j) \backslash M(h_j)$. If $M(h_j) \subseteq f_1(h_j)$, then the result is immediate. Hence, suppose that $M(h_j) \nsubseteq f_1(h_j)$. Choose any $a_s \in M(h_j) \backslash f_1(h_j)$. Clearly, either (i) $a_s$ has priority 1 but $f(a_s) = h_l \neq h_j$ or (ii) $a_s$ has priority $> 1$. In case (i), since $f(a_s) = h_l$, $a_s$ must prefer to be assigned to $h_l$ than $h_j$ by Lemma 2. Hence, if $h_l$ is undersubscribed, we can promote $a_r$ to $h_j$ and $a_s$ to $h_l$ to obtain a more popular matching, a contradiction. Thus, $h_l$ is full. By Lemma 3, $\bigcup_{p=1}^{d_l} f_p(h_l) \subseteq M(h_l)$. Since $a_s \in f_1(h_l) \backslash M(h_l)$, it follows that $d_l = 0$, i.e., $f_{1,l} > c_l$. Now, if $M(a_r) = h_l$, then we can promote $a_r$ to $h_j$ and promote $a_s$ to $h_l$ to obtain a more popular matching. Hence, $M(a_r) \neq h_l$. Choose any $a_t \in M(h_l)$. We then promote $a_r$ to $h_j$, promote $a_s$ to $h_l$ and demote $a_t$ to $l(a_t)$ to obtain a matching whose improvement in satisfaction is $w_1 + w_1 - w(a_t) > 0$. In case (ii), we can promote $a_r$ to $h_j$ and demote $a_s$ to $l(a_s)$ to obtain a more popular matching.

Hence, $d_j \geq 1$. Suppose for a contradiction that $M(h_j) \backslash \bigcup_{p=1}^{d_j} f_p(h_j) \nsubseteq f_{d_j+1}(h_j)$. It follows that there exists some agent $a_s \in M(h_j) \backslash \bigcup_{p=1}^{d_j+1} f_p(h_j)$. Recall that $a_r$ has priority $d_j+1$. Clearly, either (i) $a_s$ has priority $\leq d_j+1$ but $f(a_s) = h_l \neq h_j$, or (ii) $a_s$ has priority $> d_j + 1$. It is immediate in case (ii) that we can promote $a_r$ to $h_j$ and demote $a_s$ to $l(a_s)$ to obtain a more popular matching, a contradiction. Hence, case (i) applies. It follows by Lemma 2 that $a_s$ prefers to be assigned to $h_l$ than $h_j$, and so, if $h_l$ is undersubscribed, we can then obtain a more popular matching by promoting $a_r$ to $h_j$ and promoting $a_s$ to $h_l$. Hence $h_l$ is full. Now, if $M(a_r) = h_l$, we can then promote $a_r$ to $h_j$ and promote $a_s$ to $h_l$ to obtain a more popular matching. Hence, $M(a_r) \neq h_l$.

Let $a_s$ have priority $z_1$ so that $z_1 \leq d_j + 1$. By our definition of $f$-houses, since $h_l = f(a_s)$, if $z_1 = 1$, then $h_l$ is the first house on $a_s$'s preference list. Since $h_l$ is full, then choose any $a_t \in M(h_l)$ and let $a_t$ have priority $z_2$. We obtain an improvement in satisfaction of $w(a_r) + w(a_s) - w(a_t) = w_{d_j+1} + w_1 - w_{z_2} > 0$ by promoting $a_r$ to $h_j$, promoting $a_s$ to $h_l$ and demoting $a_t$ to $l(a_t)$. Hence, it follows that $z_1 > 1$. Then, $h_l$ must be the most preferred house on $a_s$'s preference list such that $\sum_{p=1}^{z_1-1} f_{p,l} < c_l$. By definition of $f(a_s) = h_l$, it follows that $z_1 \leq d_l + 1$. Now, by Lemma 3, $\bigcup_{p=1}^{d_l} f_p(h_l) \subseteq M(h_l)$. However, $a_s \notin M(h_l)$. Hence, it follows that $z_1 > d_l$, i.e., it follows that $z_1 = d_l + 1$. Since $\sum_{p=1}^{z_1-1} f_{p,l} < c_l$ and $h_l$ is full, it follows that $\bigcup_{p=1}^{z_1-1} f_p(h_l) \subset M(h_l)$. Hence, we have that $M(h_l) \backslash \bigcup_{p=1}^{z_1-1} f_p(h_l) \neq \emptyset$. It follows that there exists some agent $a_t \in M(h_l) \backslash \bigcup_{p=1}^{z_1-1} f_p(h_l)$. Clearly, either (i) $a_t \in \bigcup_{p=z_1}^{k} f_p(h_l)$ or (ii) $a_t \notin f(h_l)$.

Note that since $M(a_r) \neq h_l$, $a_t \neq a_r$. Now, in both case (i) and (ii), if $a_t$ has priority $z_2 \geq z_1$, we can then promote $a_r$ to $h_j$, promote $a_s$ to $h_l$ and demote $a_t$ to $l(a_t)$ to obtain a matching whose improvement in satisfaction is $w(a_r) + w(a_s) - w(a_t) = w_{d_j+1} + w_{z_1} - w(a_t) > 0$, a contradiction. Hence $z_2 < z_1$, and so only case (ii) applies. Let $h_{l'} = f(a_t)$. It is obvious, by Lemma 2, that $a_t$ prefers to be assigned to $h_{l'}$ than $h_l$. Furthermore, $h_{l'} \neq h_j$, for suppose not. As $z_2 < z_1 \leq d_j + 1$ and $f(a_t)$ is defined, it follows that $z_2 \leq d_j$. By Lemma 3, $\bigcup_{p=1}^{z_2} f_p(h_j) \subseteq \bigcup_{p=1}^{d_j} f_p(h_j) \subseteq M(h_j)$ so that $a_t \in M(h_j)$. However, this gives a contradiction since $a_t \in M(h_l)$ and $h_j \neq h_l$. Clearly also, $h_{l'} \neq M(a_r)$ for otherwise, we can promote $a_r$ to $h_j$, promote $a_s$ to $h_l$ and promote $a_t$ to $h_{l'}$ to obtain a more popular matching, a contradiction. Hence, the houses $h_{l'}$, $h_l$, $h_j$ and $M(a_r)$ are distinct. Clearly too, the agents $a_r$, $a_s$ and $a_t$ are distinct for $z_2 < z_1 \leq d_j + 1$ and $a_r \neq a_s$.

We assume that $h_{l'}$ is full, for otherwise we can obtain a contradiction by promoting $a_r$ to $h_j$, promoting $a_s$ to $h_l$ and promoting $a_t$ to $h_{l'}$. Let $a_u \in M(h_{l'})$. If $z_2 = 1$, then we can promote $a_r$ to $h_j$, promote $a_s$ to $h_l$, promote $a_t$ to $h_{l'}$ and demote $a_u$ to $l(a_u)$ to obtain a new matching with improvement in satisfaction $w(a_r) + w(a_s) + w(a_t) - w(a_u) = w_{d_j+1} + w_{z_1} + w_1 - w(a_u) > 0$. Hence, $z_2 > 1$. If we let $a_t$ and $a_u$ take the roles of $a_s$ and $a_t$ respectively, then it follows by the argument that we use to define $a_t$ that we are able to choose $a_u$ such that $a_u$ has priority $< z_2$ and $a_u \notin f(h_{l'})$. It follows that $a_u$ is an agent distinct from $a_r$, $a_s$ and $a_t$ since $P(a_u) < z_2$.

By continuing this argument, it follows that we obtain a sequence of distinct agents $a_0, a_1, a_2, a_3, ...$ where $a_0 = a_r$, $a_1 = a_s$, $a_2 = a_t$, and $a_3 = a_u$. For $i \geq 4$, the above construction indicates that $P(a_i) < P(a_{i-1})$. If this sequence does not terminate as a result of arriving at a contradiction due to any of the above cases, then we are bound to ultimately generate an agent $a_x$ such that $P(a_x) < 1$, which is impossible. □

Lemmas 3 and 4 give rise to the following corollary concerning the assignees of $f$-houses in popular matchings.

**Corollary 5.** *Let $M$ be a popular matching in any WCHA instance $I$. Then, for every $f$-house $h_j$,*

1. *if $f_j \leq c_j$, then $f(h_j) \subseteq M(h_j)$;*
2. *if $f_j > c_j$, then $|M(h_j)| = c_j$ and $\bigcup_{p=1}^{d_j} f_p(h_j) \subseteq M(h_j) \subseteq \bigcup_{p=1}^{d_j+1} f_p(h_j)$.*

*Proof.* In Case 1, if $f_j \leq c_j$, it follows by definition of $d_j$ that $\bigcup_{p=d_j+1}^{k} f_p(h_j) = \emptyset$. Clearly then, $f(h_j) = \bigcup_{p=1}^{d_j} f_p(h_j) \subseteq M(h_j)$ by Lemma 3. In Case 2, it follows by Lemmas 3 and 4 that $\bigcup_{p=1}^{d_j} f_p(h_j) \subseteq M(h_j)$, $M(h_j) \backslash \bigcup_{p=1}^{d_j} f_p(h_j) \subseteq f_{d_j+1}(h_j)$ and $|M(h_j)| = c_j$. □

We now define the concept of an *s-house* for each agent. Given a popular matching $M$, 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 $\sum_{i=1}^{z} f_{i,j} < c_j$. Note that $s(a)$ may not exist if $f(a) = l(a)$. However, all such agents will be assigned to their $f$-houses in any matching since last resort houses are unique to individual agents.

A house $h_j \in H$ is an *s-house* if $h_j = s(a)$ for some $a \in A$. To illustrate the $s$-house definition, let us look at Instance $I_1$ in Figure 1 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 $i \neq j$ as seen from this example. 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 $O(m)$.

Now, it may be shown that a popular matching $M$ will only assign an agent $a$ to either $f(a)$ or $s(a)$ as indicated by the next lemma.

**Lemma 6.** *Let $M$ be a popular matching in any WCHA instance $I$. Then, every agent $a \in A$ is assigned in $M$ to either $f(a)$ or $s(a)$.*

*Proof.* Let $a \in P_i$ and let $M(a) = h_x$. Suppose that the statement of this lemma is false. By Lemma 2, $a$ cannot be assigned to a house better than $f(a)$. Then, besides $f(a)$ or $s(a)$, $h_x$ can either be (i) a house between $f(a)$ and $s(a)$ or (ii) a house worse than $s(a)$.

In case (i), it follows that $h_x$ is an $f$-house such that $\sum_{p=1}^{i} f_{p,x} \geq c_x$, for otherwise $s(a) = h_x$. Hence, $f_x \geq c_x$ and $M(h_x) \subseteq f(h_x)$ by Corollary 5. However, $a \in M(h_x) \backslash f(h_x)$, a contradiction.

In case (ii), let $h_j = s(a)$. It follows that $a$ must prefer to be assigned to $h_j$ than $M(a) = h_x$. Clearly, $h_j$ is full, for otherwise we can promote $a$ to $h_j$, a contradiction. It follows by our definition of $s$-houses that $\sum_{p=1}^{i} f_{p,j} < c_j$. Hence, by our definition of $d_j$, $i \leq d_j$. Since $\bigcup_{p=1}^{d_j} f_p(h_j) \subseteq M(h_j)$ (by Lemma 3) and $h_j$ is full, it follows that $\bigcup_{p=1}^{i} f_p(h_j) \subset M(h_j)$ so that $M(h_j) \backslash \bigcup_{p=1}^{i} f_p(h_j) \neq \emptyset$. Hence, there exists some $a_s \in M(h_j) \backslash \bigcup_{p=1}^{i} f_p(h_j)$. It is obvious that either (i) $a_s \in \bigcup_{p=i+1}^{k} f_p(h_j)$, or (ii) $a_s \notin f(h_j)$.

Clearly in case (i), $a_s$ has priority $> i$, so we can promote $a$ to $h_j$ and demote $a_s$ to $l(a_s)$ to obtain a matching whose improvement in satisfaction is $w_i - w(a_s) > 0$. In case (ii), let $a_s$ have priority $z_1$. It follows that $z_1 \leq i$, for otherwise, we can promote $a$ to $h_j$ and demote $a_s$ to $l(a_s)$ to obtain a new matching whose improvement in satisfaction is $w_i - w_{z_1} > 0$. Let $f(a_s) = h_l$. Clearly, $a_s$ must prefer to be assigned to $h_l$ than $h_j$ by Lemma 2. If $h_l$ is undersubscribed, we can then promote $a$ to $h_j$ and promote $a_s$ to $h_l$ to obtain a more popular matching, a contradiction. Hence, suppose that $h_l$ is full. Let $a_t \in M(h_l)$.

If $z_1 = 1$, then we can promote $a$ to $h_j$, promote $a_s$ to $h_l$ and demote $a_t$ to $l(a_t)$ to obtain a matching with improvement in satisfaction $w(a) + w(a_s) - w(a_t) = w_i + w_1 - w(a_t) > 0$. Hence, suppose that $z_1 > 1$. Clearly, $h_x \neq h_l$ for suppose otherwise. By Corollary 5, $h_l$ must be an $f$-house such that $f_l > c_l$ by existence of $a_s$, for otherwise $a_s \in M(h_l)$. It follows that $M(h_l) \subseteq f(h_l)$. Now, if $h_l = h_x$, then this gives us a contradiction since $a \in M(h_l)$ but $h_x \neq f(a)$ for $a$ prefers $s(a)$ to $h_x$.

Hence, $h_l \neq h_x$. Then, $a_t \neq a$. It follows that we can reuse arguments from the proof of Lemma 4 to obtain a sequence of distinct agents $a_0, a_1, a_2, ...$ where $a_0 = a$, $a_1 = a_s$, and $a_2 = a_t$. For $j \geq 3$, the construction of the sequence indicates that $P(a_i) < P(a_{i-1})$. If this sequence does not terminate as a result of arriving at a contradiction due to any of the cases outlined in Lemma 4, then we are bound to ultimately generate an agent $a_x$ such that $P(a_x) < 1$, which is impossible. □

Corollary 5 and Lemma 6 give rise to the following result.

**Theorem 7.** *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 $|M(h_j)| = c_j$ and $\bigcup_{p=1}^{d_j} f_p(h_j) \subseteq M(h_j) \subseteq \bigcup_{p=1}^{d_j+1} f_p(h_j)$.*
2. *Every agent $a$ is assigned 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 Theorem 7. However, Theorem 7 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 1. We have at least two matchings which satisfy Conditions 1 and 2 of Theorem 7: $M_1 = \{(a_1, h_1), (a_2, h_3), (a_3, h_3), (a_4, h_5), (a_5, h_4), (a_6, h_4)\}$ 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 $M_1$ may be verified to be 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 [17] to WCHA.

### 3.1 Potential improvement paths

Let us now define a matching $M$ that satisfies Conditions 1 and 2 of Theorem 7 to be *well-formed*. Then, a PIP leading out of some $f$-house $h_0$ with respect to a well-formed matching $M$ is an alternating path $\Pi = \langle h_0, a_0, h_1, a_1, ..., h_x, a_x \rangle$ such that $h_i = f(a_i)$ and $(a_i, h_i) \in M$ for $0 \le i \le x$, and $a_i$ prefers $h_{i+1}$ to $h_i$ for $i < x$. A PIP leading out of $h_0$ always exists, which can be seen as follows. Since $h_0$ is an $f$-house and $c_0 \ge 1$, there exists some agent $a'_0 \in f(h_0) \cap M(h_0)$ by Theorem 7. Then, by definition, $\langle h_0, a'_0 \rangle$ is a PIP leading out of $h_0$. The next lemma shows that any PIP leading out of $h_0$ must contain a sequence of agents with strictly decreasing priorities. It follows that the sequence of agents in $\Pi$ must be distinct.

**Lemma 8.** *Let $M$ be a well-formed matching. Let $\Pi = \langle h_0, a_0, ..., h_x, a_x \rangle$ be a PIP with respect to $M$ leading out of $h_0$ as defined above. Then, $P(a_{i+1}) < P(a_i)$ for $0 \le i < x$.*

*Proof.* Let $a_0$ have priority $z_1$. If $x = 0$, then $a_0$ is the last (only) agent in the path. Otherwise, $x > 0$ and it follows by definition of $\Pi$ that $h_0$ is not the first house on $a_0$'s preference list as $h_1$ is a house that $a_0$ prefers to $h_0$. Hence, it must be that $h_1$ is an $f$-house such that $\sum_{p=1}^{z_1-1} f_{p,1} \ge c_1$ by definition of $f(a_0) = h_0$.

Since $M$ is well-formed and $f_1 \ge c_1$, it follows by Theorem 7 that $|M(h_1)| = c_1$ and $M(h_1) \subseteq f(h_1)$. Now, if $\sum_{p=1}^{z_1-1} f_{p,1} = c_1$, then by definition of an $f$-house, $f_{p,1} = 0$ for $z_1 \le p \le k$. Hence, $d_1 \le z_1 - 1$. Since $f_1 = c_1$, it follows that $M(h_1) \subseteq \bigcup_{p=1}^{z_1-1} f_p(h_1)$ by Theorem 7. On the other hand, if $\sum_{p=1}^{z_1-1} f_{p,1} > c_1$, then $f_1 > c_1$ and $d_1 + 1 \le z_1 - 1$. It follows by Theorem 7 again that $M(h_1) \subseteq \bigcup_{p=1}^{z_1-1} f_p(h_1)$. Clearly as a result, $M(h_1) \subseteq \bigcup_{p=1}^{z_1-1} f_p(h_1)$ in all cases.

Since $a_1 \in M(h_1)$, it follows that $f(a_1) = h_1$ and $a_1$ has priority strictly less than $z_1$. Moreover, we can repeat the argument to deduce the priority of each agent $a_i$ in $\Pi$. It is then straightforward to see that the priority of any agent in $\Pi$ must be strictly less than its predecessor so that $P(a_{i+1}) < P(a_i)$ for each $i \ge 0$. □

Let us define the cost of $\Pi$ to be $cost(\Pi) = w(a_x) - w(a_{x-1}) - ... - w(a_0)$ if $x > 0$. Note that $cost(\Pi) = w(a_0)$ 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 $h_0$ to $M(a_r)$. The next lemma shows that any such agent cannot belong to $\Pi$. 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 $a_i$ to $h_{i+1}$ for all $i < x$ and demote $a_x$ to $l(a_x)$ to obtain a new matching that is more popular than $M$.

**Lemma 9.** *Let $M$ be a well-formed matching. Let $\Pi = \langle h_0, a_0, ..., h_x, a_x \rangle$ be a PIP with respect to $M$ leading out of $h_0$ as defined above. Then, any agent $a$ who prefers $h_0$ to $M(a)$ does not belong to $\Pi$.*

**Algorithm 2** First stage of Algorithm Prune-WCHA

**for each** $f$-house $h$ **do**<br>
  $\lambda(h) := w_1$;  // a suitable upper bound<br>
**end for**<br>
**for** $z$ in $1..k$ **do**<br>
  **for each** $a \in P_z$ **do**<br>
    let $S$ contain the set of houses that $a$ prefers to $f(a)$;<br>
    **if** $S \neq \emptyset$ **then**<br>
      $\lambda_{min}(a, f(a)) := \min\{\lambda(h) : h \in S\}$;<br>
    **else**<br>
      $\lambda_{min}(a, f(a)) := \infty$;  // a suitable default value<br>
    **end if**<br>
    **if** $\lambda_{min}(a, f(a)) < w_z$ **then**<br>
      **return** "No popular matching exists";<br>
    **end if**<br>
  **end for**<br>
  **for each** $f_z$-house $h_j$ **do**<br>
    $f'_z(h_j) := f_z(h_j)$;<br>
    **if** $z \leq d_j$ **then**<br>
      **for each** $a \in f'_z(h_j)$ **do**<br>
        remove $(a, s(a))$ from $G'$;<br>
      **end for**<br>
    **else** // $z = d_j + 1$<br>
      **for each** $a \in f'_z(h_j)$ such that $\lambda_{min}(a, h_j) < 2w_z$ **do**<br>
        remove $(a, h_j)$ from $G'$;<br>
        remove $a$ from $f'_z(h_j)$;<br>
      **end for**<br>
      **if** $f'_z(h_j) = \emptyset$ **then** // $|f'_z(h_j)| < c_j - \sum_{p=1}^{d_j} f_{p,j}$<br>
        **return** "No popular matching exists.";<br>
      **end if**<br>
    **end if**<br>
    $\lambda_z(h_j) := \min(w_z, \min\{\lambda_{min}(a, h_j) - w_z : a \in f'_z(h_j)\})$;  // $\lambda_{min}(a, h_j) \geq w_z$<br>
    $\lambda(h_j) := \min(\lambda(h_j), \lambda_z(h_j))$;<br>
    **if** $z > d_j$ **and** $\lambda(h_j) < w_z$ **then**<br>
      **return** "No popular matching exists.";<br>
    **end if**<br>
  **end for**<br>
**end for**

*Proof.* Let $a$ have priority $z$. Since $M$ is well-formed, either (i) $M(a) = f(a)$ or (ii) $M(a) = s(a)$. It follows in case (i) that $\sum_{p=1}^{z-1} f_{p,0} \geq c_0$ by definition of $f(a)$. In case (ii), either (a) $h_0 = f(a)$ or (b) $h_0$ is an $f$-house such that $h_0 \neq f(a)$ and $\sum_{p=1}^{z} f_{p,0} \geq c_0$ by definition of $s(a)$. Now, in subcase (a), if $\sum_{p=1}^{z} f_{p,0} < c_0$, then $z \leq d_0$ so that $\bigcup_{p=1}^{z} f_p(h_0) \subseteq \bigcup_{p=1}^{d_0} f_p(h_0) \subseteq M(h_0)$ since $M$ is a well-formed matching. However, this implies that $a \in M(h_0)$, a contradiction. It follows in all cases that $\sum_{p=1}^{z} f_{p,0} \geq c_0$. Using a similar argument as in Lemma 8, we can establish that $|M(h_0)| = c_0$ and $M(h_0) \subseteq \bigcup_{p=1}^{z} f_p(h_0)$. It follows that $P(a) \geq P(a_0)$ and hence, the priority of $a$ must be greater than the priority of any other agent in $\Pi$ by Lemma 8. Since $a \neq a_0$, $a$ cannot be an agent in $\Pi$. □

## 3.2 Pruning the graph

Let us now introduce Algorithm Prune-WCHA 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, with the first stage shown in Algorithm 2 and the second stage shown in

**Algorithm 3** Second stage of Algorithm Prune-WCHA

**for each** $a \in A$ **do**
  let $h_l := s(a)$;
  let $R$ contain the set of houses that $a$ prefers to $h_l$;
  let $S$ contain the set of houses that $a$ prefers to $f(a)$;
  $R := R - (S \cup \{f(a)\})$;
  **if** $R \neq \emptyset$ **then**
    $\lambda_{min}(a, h_l) := \min \{\lambda(h) : h \in R\}$;
  **else**
    $\lambda_{min}(a, h_l) := \infty$;   // a suitable default value
  **end if**
  **if** $\lambda_{min}(a, h_l) < w(a)$ **or** $f_l \geq c_l$ **then**
    remove $(a, h_l)$ from $G'$;
  **end if**
**end for**

Algorithm 3. 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 leading out of 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 edge removals) – we refer to $G''$ as the *pruned graph*. 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 $\lambda_z(h_j)$ as a variable and its value at the end of the algorithm equals 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 $\lambda_z(h_j)$. Note that we initialise $\lambda(h)$ to $w_1$ for every $f$-house $h$ at the outset of the first stage of Algorithm Prune-WCHA, for if $\Pi$ is any PIP leading out of $h$, then $cost(\Pi) \leq w(a_x)$, where $a_x$ is the final agent on the path. However, $w(a_x) \leq w_1$. Hence, $w_1$ is an upper bound for the final computed value of $\lambda(h)$. 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, as we shall show, 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 $f(a_s)$. Note that $S$ will be empty if $f(a_s)$ is the first house on $a_s$'s preference list. 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, the algorithm sets $\lambda_{min}(a_s, f(a_s))$ to $\infty$ as a suitable default value. 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, the algorithm sets $\lambda_{min}(a_s, s(a_s))$ to $\infty$ as a suitable default value.

### 3.3 Proof of correctness

The following lemma gives an important technical result regarding the correctness of the algorithm.

**Lemma 10.** *Let $z$ be an iteration of the for loop on line 4 of the first stage of Algorithm Prune-WCHA. Suppose that, by the end of this iteration, the algorithm has not terminated with a report that no popular matching exists. Let $h_j \in H$ be any $f_z$-house. Then, at the end of this loop iteration:*

1. *for each $a \in P_z$, if $f(a)$ is not the first ranked house in a's preference list, then $\lambda_{min}(a, f(a))$ equals the minimum cost of all PIPs among all houses that a prefers to $f(a)$ taken over all well-formed matchings in $G''$; else, $\lambda_{min}(a, f(a)) = \infty$.*
2. *$\lambda_z(h_j)$ stores the minimum cost among all PIPs taken over all well-formed matchings in $G''$ such that $(a, h_j)$ is the first edge for some $a \in P_z$.*
3. *$\lambda(h_j)$ stores the minimum cost among all PIPs taken over all well-formed matchings in $G''$ such that $(a, h_j)$ is the first edge for some $a \in P_q$ where $1 \leq q \leq z$.*
4. *if any edge has been removed from $G'$, then it cannot be part of any popular matching.*

*Proof.* Given $1 \leq z \leq k$, we will proceed by induction on $z$.

For the base case, let $z = 1$. If $a \in P_1$, then clearly $S = \emptyset$ for $a$ so that $\infty$ is assigned to $\lambda_{min}(a, f(a))$ as required in line 10. Now, any PIP leading out of $h_j$ and containing the edge $(a, h_j)$ ends at $a$ and has cost $w_1$. Clearly, $w_1$ is assigned to $\lambda_z(h_j)$ as required at line 31 since $\lambda_{min}(a', h_j) = \infty$ for each $a' \in f'_1(h_j)$. Also, $w_1$ is assigned to $\lambda(h_j)$ at line 32 as required, since this is the minimum of $\lambda_z(h_j)$ and the initialised value of $\lambda(h_j)$ which is also $w_1$. Finally, the only edges removed during this iteration are dealt with at lines 19-21 (as the condition in line 23 is not satisfied). For, clearly if $a \in P_1$ and $d_j \geq 1$, $a$ must be assigned to $f(a) = h_j$ and not $s(a)$ in any well-formed matching $M$ by Condition 1 of Theorem 7. Hence, the edge $(a, s(a))$ cannot belong to any popular matching.

For the inductive case, let us assume that $2 \leq z \leq k$, and that the result is true for $z - 1$. Let $a \in P_z$ be any agent. Suppose that $S \neq \emptyset$. Choose any $h_l \in S$. It follows that $\sum_{p=1}^{z-1} f_{p,l} \geq c_l$ by definition of $h_j = f(a)$. Hence, it is impossible that $h_l$ can be an $f_p$-house for any $p \geq z$. By the inductive hypothesis, $\lambda(h_l)$ stores the minimum cost among all PIPs leading out of $h_l$ where $(a', h_l)$ is the first edge for some $a' \in P_q$ where $1 \leq q \leq z - 1$. Hence, $\lambda(h_l)$ stores the minimum cost among all PIPs leading out of $h_l$ at the end of the iteration $z - 1$. Thus, if $S \neq \emptyset$, then when $\lambda_{min}(a, f(a))$ is defined during iteration $z$ in line 8, it contains the minimum cost of a PIP leading out of any house that $a$ prefers to $f(a)$; otherwise, $S = \emptyset$ and $\lambda_{min}(a, f(a))$ is assigned to be $\infty$ in line 10 as required.

Now, it follows that the minimum cost of a PIP out of $h_j$ for which the first edge is $(a, h_j)$ such that $a \in f_z(h_j)$ either stops at $a$ and has cost $w_z$, or it continues. If it continues, it must do so with some edge $(a, h_l)$ such that $a$ prefers $h_l$ to $h_j$. Hence, the minimum cost of a PIP out of $h_j$ for which the first edge is $(a, h_j)$ is the minimum of $w_z$ and $\lambda_{min}(a, h_j) - w_z$. Clearly then, this is exactly the value assigned to $\lambda_z(h_j)$ on line 31 as required. Also, it follows by the inductive hypothesis that $\lambda(h_j)$ should be set at iteration $z$ to be the minimum of $\lambda_z(h_j)$ and the value of $\lambda(h_j)$ at the end of iteration $z - 1$. This is precisely the value assigned to $\lambda(h_j)$ at line 32.

Finally, it remains to show that any edge removed during iteration $z$ cannot belong to part of any popular matching. Now, if $z \leq d_j$, then it follows by Theorem 7 that $a$ must be assigned to $h_j$ and not $s(a)$ for any well-formed matching $M$. Hence, the edge $(a, s(a))$

cannot belong to any well-formed matching and is deleted in line 20 as required. Clearly, if $f_j \le c_j$, then it is bound to be the case that $z \le d_j$.

On the other hand, if $z > d_j$, then it follows that in any well-formed matching $M$, $\bigcup_{p=1}^{d_j} f_p(h_j) \subseteq M(h_j)$ but only a proper subset of $f_{d_j+1}(h_j)$ will be assigned to $h_j$ in $M$. Now, suppose that $a \in M(h_j) \cap f_{d_j+1}(h_j)$. It follows that $z = d_j + 1$. Let $h_l$ be any house that $a$ prefers to $h_j$, supposing that such a house exists. Clearly, if there exists a minimum cost PIP $\Pi$ out of $h_l$ such that $cost(\Pi_{min}(h_l)) - w_z < w_z$, then $\Pi$ can be used to promote $a$ to $h_l$, and in the process, free up a space in $h_j$ which can thus be assigned to any agent $a'$ in $f_{d_j+1}(h_j) \backslash M(h_j)$. Clearly, $M(a') = s(a')$ since $M$ is well-formed so that $a'$ improves as result. It follows that $M$ cannot be popular since we can promote $a'$ to $h_j$, promote $a$ to $h_l$ and promote along $\Pi$ to obtain a more popular matching than $M$. Hence, if $\lambda_{min}(a, h_j) < 2w_z$, then $M$ is not popular. Since $M$ is arbitrary, the edge $(a, h_j)$ cannot belong to any popular matching so that we delete it in line 24.

Note that $\Pi_{min}(h_l)$ must be a minimum cost PIP with respect to $M$. For, let us consider the first edge $(b, h_l)$ in $\Pi_{min}(h_l)$. Note that $f_l \ge c_l$ and $d_l + 1 < z$ since $h_l$ is a house that $a$ prefers to $f(a) = h_j$.

Suppose firstly that $b \in f_{d_l+1}(h_l)$. Let $\lambda_{d_l}$ be the value of $\lambda(h_l)$ at the end of phase $d_l$. Now, we have that the value of $\lambda(h_l)$ as computed in phase $d_l + 1$ by lines 31-32 of the algorithm is equal to $\min(w_{d_l+1}, \lambda_{d_l}, \min\left\{\lambda_{min}(b', h_l) - w_{d_l+1} : b' \in f'_{d_l+1}(h_l)\right\})$. Let us suppose that $\min\left\{\lambda_{min}(b', h_l) - w_{d_l+1} : b' \in f'_{d_l+1}(h_l)\right\} < w_{d_l+1}$. Then, there exists some agent $b' \in f'_{d_l+1}(h_l)$ such that $\lambda_{min}(b', h_l) - w_{d_l+1} < w_{d_l+1}$, i.e., $\lambda_{min}(b', h_l) < 2w_{d_l+1}$. However, such a $b'$ would have been removed from $f'_{d_l+1}(h_l)$ at line 25, a contradiction. Hence, $\lambda_{min}(b', h_l) - w_{d_l+1} \ge w_{d_l+1}$ for all $b' \in f'_{d_l+1}(h_l)$. It follows that any minimum cost PIP in $G''$ (with respect to any well-formed matching) with $(b', h_l)$ as its first edge must have cost greater than or equal to $w_{d_l+1}$, i.e., $cost(\Pi_{min}(h_l)) \ge w_{d_l+1}$. Now, suppose that $\lambda_{d_l} < w_{d_l+1}$. Then, there exists a PIP leading out of $h_l$ whose first edge is $(c, h_l)$ where $P(c) \le d_l$, with cost less than $w_{d_l+1}$. However, this then contradicts the fact that the PIP with $(b, h_l)$ as its first edge has minimum cost for $h_l$ as we supposed. Hence, $w_{d_l+1}$ is a lower bound for the final computed value of $\lambda(h_l)$. Clearly then, $\lambda(h_l) = w_{d_l+1}$. Since $(b, h_l)$ is the first edge of $\Pi_{min}(h_l)$ where $b \in f_{d_l+1}(h_l)$, then as this path is defined with respect to some well-formed matching, it follows that $(b', h_l) \in M$ for some $b' \in f_{d_l+1}(h_l)$ (possibly $b = b'$), since $M$ is well-formed. Then, $\langle h_l, b' \rangle$ is a PIP of cost $w_{d_l+1}$ with respect to $M$. Moreover, since $w_{d_l+1} = cost(\Pi_{min}(h_l)) < 2w_z$ as established in the previous paragraph, it follows that we can promote $a$ to $h_l$, promote $a'$ to $h_j$ and demote $b'$ from $h_l$ so that $M$ is not popular as shown above.

Hence, $b \in \bigcup_{p=1}^{d_l} f_p(h_l)$. Clearly then, $(b, h_l)$ must belong to every well-formed matching by Condition 1(a) of Theorem 7 so that $(b, h_l)$ must belong to $M$. It follows that we can repeat the above argument to show that $\Pi_{min}(h_l)$ is a minimum cost PIP with respect to $M$ by considering the remaining alternate edges in $\Pi_{min}(h_l)$. If each alternate edge $(c, h_x)$ satisfies the condition $c \in \bigcup_{p=1}^{d_x} f_p(h_x)$, then the result is immediate. Otherwise, it must be the case that we encounter some edge $(c', h_{x'})$ in $\Pi_{min}(h_l)$ such that $c' \in f_{d_{x'}+1}(h_{x'})$. Clearly then, $(c', h_{x'})$ is the final edge in $\Pi_{min}(h_l)$ so that we must be able to promote $a$ to $h_l$, promote $a'$ to $h_j$ and promote along $\Pi_{min}(h_l)$ to obtain a more popular matching than $M$ by a similar argument to that in the previous paragraph. □

The next three lemmas establish the correctness of the algorithm.

**Lemma 11.** *Suppose that Algorithm Prune-WCHA does not terminate during the execution of its first stage by reporting that no popular matching exists. Then, any edge removed by Algorithm Prune-WCHA over both stages cannot belong to a popular matching.*

*Proof.* By Lemma 10, any edges removed by Algorithm Prune-WCHA in the first stage cannot belong to any popular matching. We now show that any edges removed by the algorithm in the second stage also cannot belong to any popular matching.

Let $M$ be any well-formed matching. Let $a$ be any agent and let $P(a) = z$. Also, let $R$ contain the set of houses between $f(a)$ and $s(a)$ on $a$'s preference list that $a$ prefers to $s(a)$ (not including $f(a)$ and $s(a)$). Let $s(a) = h_l$. Suppose that $M(a) = h_l$. Let $h_j \in R$ and suppose that $cost(\Pi_{min}(h_j)) < w_z$. Clearly, $\Pi_{min}(h_j)$ must be a minimum cost PIP with respect to $M$ by a similar argument to that used in the proof of Lemma 10. Then, $\Pi_{min}(h_j)$ can be used to free up $h_j$ and promote $a$ to $h_j$ to obtain a more popular matching than $M$. Hence, $M$ cannot be popular. It follows that an edge pruned due to the first condition in line 11 of the second stage of the algorithm cannot belong to any popular matching.

Now, if $f_l \geq c_l$ and $M(a) = h_l$, then $M$ cannot be popular by Condition 1 of Theorem 7, since $M(h_l) \not\subseteq \bigcup_{p=1}^{d_l+1} f_p(h_l)$. This shows that the edge $(a, h_l)$ pruned due to the second condition in line 11 of the second stage of the algorithm also cannot belong to any popular matching.

It thus follows that any edges removed by the algorithm cannot belong to a popular matching. □

**Lemma 12.** *If Algorithm Prune-WCHA reports that no popular matching exists, then $I$ does not admit a popular matching.*

*Proof.* Let us consider the cases where Algorithm Prune-WCHA reports that no popular matchings exist as a result of some condition being satisfied: (i) lines 12-13, (ii) lines 27-28 and (iii) lines 33-34 during some iteration $z$ of the for loop on line 4 of the first stage. Suppose for a contradiction that $M$ is a popular matching in $I$. Then $M$ is a well-formed matching in $G'$ by Theorem 7. Also $M$ is a well-formed matching in $G''$, since no edge of $M$ is deleted by the algorithm up to this point by Lemma 10.

In case (i), let $a$ be the agent considered during the relevant iteration of the for loop on line 5 when the algorithm terminates. Then $P(a) = z$. Let $h_j = f(a)$ and let $h_l$ be a house that $a$ prefers to $h_j$ such that $\lambda_{min}(a, h_j) = cost(\Pi_{min}(h_l))$. It follows by a similar argument to that used in the proof of Lemma 10 that $\Pi_{min}(h_l)$ must be a minimum cost PIP with respect to $M$. Now, if $\lambda(h_l) < w_z$, then we can use $\Pi_{min}(h_l)$ to free $h_l$ and then promote $a$ to $h_l$ to obtain a more popular matching than $M$. Hence, $M$ cannot be popular, a contradiction.

In case (ii), clearly $f_j > c_j$. Now, if $f'_{d_j+1}(h_j) = \emptyset$ after the removal of edges in lines 23-26, then it follows that no well-formed matching can exist in $G''$ since no matching can satisfy Condition 1(b) of Theorem 7, a contradiction to the earlier observation that $M$ is a well-formed matching in $G''$.

In case (iii), $z = d_j + 1$. Clearly, only a proper subset of agents in $f_{d_j+1}(h_j)$ can be assigned to $h_j$ in $M$ since $f_j > c_j$. Let $a \in f_{d_j+1}(h_j) \backslash M(h_j)$. Note that $\Pi_{min}(h_j)$ must be a minimum cost PIP with respect to $M$ using a similar argument in the proof of Lemma 10. Now, if $\lambda(h_j) < w_{d_j+1}$, then $\Pi_{min}(h_j)$ can be used to free up a place in $h_j$ and then promote $a$ (who must be assigned to $s(a)$ in $M$) to $h_j$ to obtain a matching that is more popular than $M$, a contradiction. □

**Lemma 13.** *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.*

*Proof.* Now, if $M$ is not popular, it follows that there exists another matching $M'$ which is more popular than $M$. Let us clone $G''$ to obtain a cloned graph $C(G'')$ as follows. We replace every house $h_j \in H$ with the clones $h_j^1, h_j^2, \ldots, h_j^{c_j}$. We then divide the capacity of each house among its clones by allowing each clone to have capacity 1. In addition, if $(a, h_j)$ is an edge in $G''$, then we add $(a, h_j^p)$ to the edge set of $C(G'')$ for all $p$ $(1 \leq p \leq c_j)$. Let us then adapt the well-formed matching $M$ in $G''$ to obtain its clone $C(M)$ in $C(G'')$ as follows. If a house $h_j$ in $G''$ is assigned to $x_j$ agents $a_1, ...a_{x_j}$ in $M$, then we add $(a_p, h_j^p)$ to $C(M)$ for $1 \leq p \leq x_j$, so that $|C(M)| = |M|$. We repeat a similar process for $M'$ to obtain its clone $C(M')$ in $C(G'')$.

Let us consider $X = C(M) \oplus C(M')$. Since $sat(M', M) > 0$, let $a \in A$ be an agent who prefers $M'$ to $M$. Let $P(a) = z$ and let $M'(a) = h_j$. We will show that there exists a PIP $\Pi$ leading out of $h_j$ with respect to $M$. Since $M$ is well-formed, we can reuse a similar argument to the proof of Lemma 9 to establish that $h_j$ is an $f$-house such that $\sum_{p=1}^{z} f_{p,j} \geq c_j$. It follows that $h_j$ is full in $M$ and $M(h_j) \subseteq f(h_j)$ by Theorem 7. Let $a_r \in M(h_j) \backslash M'(h_j)$ ($a_r$ must exist since $h_j$ is full in $M$) and let $P(a_r) = z_1$. Then, $a \neq a_r$. Also, it follows that $f(a_r) = h_j$ and $z_1 \leq z$. If $a_r$ does not prefer $M'$ to $M$, then we finish tracing $\Pi$. Otherwise, we will extend $\Pi$ to make sure that it ends with some agent $b$ who prefers $M$ to $M'$. It follows by definition of $f(a_r)$ that $M'(a_r) = h_l$ is an $f$-house that $a_r$ prefers to $h_j$ such that $\sum_{p=1}^{z_1-1} f_{p,l} \geq c_l$ and hence by Theorem 7, $M(h_l) \subseteq f(h_l)$. Let $a_s \in M(h_l) \backslash M'(h_l)$ and let $P(a_s) = z_2$. Clearly then, $z_2 < z_1$. It follows by the same argument as for $a_r$ that if $a_s$ does not prefer $M'$ to $M$, then we finish tracing $\Pi$, i.e., $\Pi = \langle h_j, a_r, h_l, a_s \rangle$. Otherwise, we repeat the argument until we encounter an agent $a_t$ who does not prefer $M'$ to $M$ so that $\Pi$ terminates. Clearly, this will eventually happen since all agents in $\Pi$ are assigned in $M$ to their $f$-house and the priority levels of agents are strictly decreasing so that we must eventually reach some agent $a_t \in P_1$ such that $M(a_t) = f(a_t)$. However, it is then impossible that $a_t$ prefers $M'$ to $M$. Finally, by construction of $\Pi$, it follows that $\Pi$ belongs to $X$ since $\Pi$ (with appropriate superscripts for house clones) consists of alternate edges in $C(M) \backslash C(M')$ and $C(M') \backslash C(M)$.

We have established that for every $a \in P(M', M)$, there exists a PIP $\Pi(a)$ leading out of $h_j$, where $h_j = M'(a)$. Let $\Gamma = \{\Pi(a) : a \in P(M', M)\}$ and let $\Gamma' \subseteq \Gamma$ contain only those maximal PIPs in $\Gamma$. We will show that there exists an agent $d \in A$ such that $\Pi(d) \in \Gamma'$ and $cost(\Pi(d)) < w(d)$. For, suppose that $cost(\Pi(a)) \geq w(a)$ for every $\Pi(a) \in \Gamma'$. Let $\Pi(a) \in \Gamma'$ and let $\Pi(a) = \langle h_0, a_0, h_1, a_1, ..., h_x, a_x \rangle$. We define $l(\Pi(a)) = a_x$. Also, $cost(\Pi(a)) = w(a_x) - w(a_{x-1}) - ... - w(a_0) \geq w(a)$, i.e., $w(a) + w(a_0) + ... + w(a_{x-1}) \leq w(a_x)$. Now, $\{a, a_0, ..., a_{x-1}\} \subseteq P(M', M)$ whilst $a_x \in P(M, M')$. Let $D$ be the connected component of $X$ containing $\Pi(a)$ (with appropriate superscripts for house clones). It follows that $D$ must be a path or cycle whose edges alternate between $C(M)$ and $C(M')$. Clearly, $D$ cannot be an even-length alternating path with more agents than houses, or an odd-length alternating path whose end edges belong to $C(M')$, for otherwise we have an agent who is unassigned in $C(M)$ and hence in $M$, a contradiction to the definition of a well-formed matching. Hence, $D$ is either an (i) even-length alternating path with more houses than agents, or (ii) an odd-length alternating path whose end edges belong to $C(M)$, or (iii) a cycle. It is obvious that $D$ contains distinct agents and so we cannot have overlapping maximal PIPs. Hence, by construction of $\Gamma'$, the agents in $\Pi(a)$, together with $a$, but not including $l(\Pi(a))$, taken over all $\Pi(a) \in \Gamma'$, form a partition of $P(M', M)$. Moreover, for every such $a$, we have established the existence of some $l(\Pi(a)) \in P(M, M')$. Hence,

$$\begin{aligned}\sum_{a\in P(M',M)} w(a) &= \sum_{\Pi(a)\in\Gamma'} w(a) + \sum_{\Pi(a)\in\Gamma'} \sum \big\{w(a') : a' \in \Pi(a) \wedge a' \neq l(\Pi(a)\big\} \\ &\leq \sum_{\Pi(a)\in\Gamma'} \big\{w(a') : a' = l(\Pi(a))\big\} \\ &\leq \sum_{a\in P(M,M')} w(a)\end{aligned}$$

It follows that $sat(M', M) \leq 0$, a contradiction. As a result, $cost(\Pi(d)) < w(d)$ for some $\Pi(d) \in \Gamma'$. Let $h_j = M'(d)$. Now, if $M(d) = f(d)$, then lines 12-13 of the first stage of the algorithm would report that no popular matching exists since $\lambda_{min}(d, f(d)) < w(d)$, a contradiction. Hence, $M(d) = s(d)$ and $h_j$ is (i) better than $f(d)$, or (ii) equal to $f(d)$, or (iii) between $f(d)$ and $s(d)$ on $a$'s preference list. In case (i), we obtain the same contradiction as when $M(d) = f(d)$ since $\lambda_{min}(d, f(d)) < w(d)$. In case (ii), $f(d) = h_j$. Since $M(d) = s(d)$, it must be the case that $d \in f_{d_j+1}(h_j)$ for otherwise $(d, s(d))$ would have been deleted by line 20 of the first stage of the algorithm. Clearly though, lines 33-34 of the first stage of the algorithm would report that no popular matching exists, a contradiction. In case (iii), $(d, s(d))$ would have been deleted by lines 11-12 of the second stage of the algorithm since $\lambda_{min}(d, s(d)) < w(d)$, a contradiction. It follows that we obtain a contradiction in all cases so that $M'$ is not more popular than $M$. □

Finally, the next lemma shows that if there is no well-formed matching in the pruned graph $G''$, then no popular matching exists.

**Lemma 14.** *Let $G''$ be the pruned graph for a given WCHA instance $I$. If there is no well-formed matching in $G''$, then no popular matching exists in $I$.*

*Proof.* Suppose that there exists a popular matching $M$ in $I$. Now, by Theorem 7, $M$ is a well-formed matching in $G'$. Moreover, all edges of $M$ must belong to $G''$ by Lemma 11. However, this implies that $M$ is a well-formed matching in $G''$, a contradiction. □

We now use the example in Figure 1 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 of the first stage, and $(a_2, h_4)$ and $(a_3, h_5)$ in phase 2 of the first stage (all in line 20 of the first stage) 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 24-25 of the first stage) since $\lambda_{min}(a_4, h_4) = 3 < 2w(a_4)$. No further edges are removed in the second stage.

## 3.4 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.

### Algorithm Popular-CHA

We give a brief recap of Algorithm Popular-CHA, shown in Algorithm 4, for finding a popular matching or reporting that none exists, given an instance $I$ of CHA [14]. For consistency with [14], the algorithm pseudocode and the accompanying description in this subsection assumes that Algorithm Popular-CHA will be applied to $G'$, however when we

**Algorithm 4** Algorithm Popular-CHA

```
 1: M := ∅;
 2: for each f-house h_j do
 3:    c'_j := c_j;
 4:    if f_j ≤ c_j then
 5:       for each a_i ∈ f(h_j) do
 6:          M := M ∪ {(a_i, h_j)};
 7:          delete a_i and its incident edges from G';
 8:       end for
 9:       c'_j := c_j − f_j;
10:    end if
11: end for
12: remove all isolated and full houses, and their incident edges, from G';
13: compute a maximum matching M' in G' using capacities c'_j;
14: if M' is not agent-complete in G' then
15:    output "no popular matching exists";
16: else
17:    M := M ∪ M';
18:    for each a_i ∈ A do
19:       h_j := f(a_i);
20:       if f_j > c_j and |M(h_j)| < c_j and h_j ≠ M(a_i) then
21:          promote a_i from M(a_i) to h_j in M;
22:       end if
23:    end for
24: end if
```

return to the WCHA context in the next subsection, this algorithm will in fact be applied to $G''$. The algorithm begins by using a pre-processing step (lines 2-12) on $G'$ that matches each agent to their first-choice house $h_j$ whenever $f_j \leq c_j$, so as to satisfy Condition 1(a) of the following theorem, which is a counterpart of Theorem 7 for CHA:

**Theorem 15** ([14]). *A matching $M$ is popular in a CHA instance $I$ if and only if*

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 $|M(h_j)| = c_j$ and $M(h_j) \subseteq f(h_j)$.*
2. *$M$ is an* agent-complete *matching (i.e., a matching in which all agents are assigned) in $G'$.*

The next step of Algorithm 4 computes a maximum matching $M'$ in $G'$, according to the adjusted house capacities $c'_j$ that are defined following pre-processing. The subgraph $G'$ can be viewed as an instance of the Upper Degree-Constrained Subgraph problem (UDCS) [7]. (An instance of UDCS is essentially the same as an instance of CHA, except that agents have no explicit preferences in the UDCS case; the definition of a matching is unchanged.) We use Gabow's algorithm [7] to compute $M'$ in $G'$ and then test whether $M'$ is agent-complete. The pre-allocations are then added to $M'$ to give $M$. As a last step, we ensure that $M$ also meets Condition 1(b) of Theorem 15. For, suppose that $h_j \in H$ is an $f$-house such that $f_j > c_j$. Then by definition, $h_j$ cannot be an $s$-house. Thus if $a_k \in M(h_j)$ prior to the third for loop, it follows that $a_k \in f(h_j)$. At this stage, if $h_j$ is undersubscribed in $M$, we repeatedly promote any agent $a_i \in f(h_j) \backslash M(h_j)$ from $M(a_i)$ (note that $M(a_i)$ must be $s(a_i)$ and hence cannot be an $f$-house $h_l$ such that $f_l > c_l$) to $h_j$ until $h_j$ is full, ensuring that $M(h_j) \subseteq f(h_j)$.

### Using Algorithm Popular-CHA for WCHA

We now show how to use Algorithm Popular-CHA in order to find a popular matching or report that none exists, given an instance of WCHA. Firstly we consider the problem of trying to assign agents to each $f$-house $h_j$ so that $h_j$ satisfies Condition 1 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 15. This work is done by lines 2-11 of Algorithm Popular-CHA. On the other hand, if $f_j > c_j$, we need to ensure that those agents with priority at most $d_j$ are assigned 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 20 in the first stage of Algorithm Prune-WCHA ensures the removal of the edge $(a, s(a))$ for every $a \in \bigcup_{p=1}^{d_j} f_p(h_j)$, it follows that $a$ must be assigned to $f(a)$ if an agent-complete matching is to exist. This is equivalent to the work done by lines 13-15 of Algorithm Popular-CHA, which tries to find an agent-complete matching and reports that no popular matching exists if unsuccessful. Furthermore, lines 18-23 of Algorithm Popular-CHA also ensure that if $f_j > c_j$, then $|M(h_j)| = c_j$ and $M(h_j) \backslash \bigcup_{p=1}^{d_j} f_p(h_j) \subseteq f_{d_j+1}(h_j)$. Lastly, we need to ensure that each agent is assigned to either $f(a)$ or $s(a)$ and it is evident that running Algorithm Popular-CHA on the pruned graph $G''$ does this. Hence, we can find a popular matching in WCHA, if one exists, by running Algorithm Popular-CHA on $G''$. As illustration, if we run Algorithm Popular-CHA on the example in Figure 1 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.

### Analysis of the algorithm for WCHA

Let us now consider the time taken to find a popular matching or to report that no such matching exists, given an instance of WCHA. First of all, it takes $O(m)$ time to define the $f$- and $s$-houses, as discussed in Section 2. Let us then consider the time complexity of Algorithm Prune-WCHA. It is clear that the subgraph $G'$ can be constructed in $O(m)$ time and has $O(n_1)$ edges since each agent has degree 2 in $G'$. Clearly, in the first stage of the algorithm, initialising $\lambda(h_j)$ for each $f$-house takes $O(n_2)$ time. Next, we iterate over every agent $a$ to define $\lambda_{min}(a, f(a))$. In order to do so, we traverse the preference list of $a$ to find the minimum cost of all PIPs among all houses that $a$ prefers to $f(a)$, if such houses exist. Even though this occurs in phases, with the total number of phases equal to the number of priority levels, the computation time for this is bounded by the total length of the preference lists. Hence, defining $\lambda_{min}(a, f(a))$ for every agent $a$ takes $O(m)$ time overall.

In order to define $\lambda_z(h_j)$ (and hence $\lambda(h_j)$) for each $f$-house $h_j$, we need to iterate over every agent $a$ such that $a \in f_z(h_j)$. Again, the time complexity for this is bounded by the total length of preference lists so that it takes $O(m)$ time overall to define $\lambda_z(h_j)$ (and hence $\lambda(h_j)$) for each $f$-house and to remove those edges which cannot belong to any popular matching (in lines 20 and 24-25 of the first stage of the algorithm). By a similar argument, the second stage of the algorithm also takes $O(m)$ time so that Algorithm Prune-WCHA takes $O(m)$ time overall. 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. It follows that we obtain the following results for the time complexity of finding a popular matching in WCHA.

**Theorem 16.** *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.*

### 3.5 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 [14] for the analogous problem in CHA 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 assigned to their last resort house. We let $A'$ denote the set $\left\{a \in \bigcup_{p=1}^{d_j} f_p(h_j) : h_j \in H\right\}$. We begin by carrying out a pre-processing step on $G''$ to compute a matching $M_0$ that assigns 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 and their incident edges. If $M'$ is not an agent-complete matching of $A_2\backslash A'$-agents, 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 assign additional $A_1\backslash A'$-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 in [14]. Let $M''$ be the matching obtained by augmenting $M'$. If any $A_1$-agent remains unassigned 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 assigned to his $f$-house $h_j$ but $h_j$ is undersubscribed in $M$, we promote $a$ from $M(a)$ to $h_j$, repeating this process as necessary. Then, clearly the matching $M$ obtained will be a well-formed matching in $G''$, and hence popular by Lemma 13. It follows that $M$ is a maximum popular matching, giving the following theorem.

**Theorem 17.** *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.*

### 3.6 "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 WHAT, and then to apply the $O(\min(k\sqrt{n}, n)m)$ algorithm of [17] 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 [17] 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 $\Omega(\sqrt{n_1 + C}c_{min})$. It follows that the cloning method is slower than our direct approach for all possible cases.

## 4 Open problem

We conclude with the following open problem. Suppose that we are presented with an instance $J$ of WCHA in which the preference lists of agents are allowed to contain ties, i.e., an instance of WCHAT. Is the problem of finding a popular matching (or reporting that none exists) in $J$ then solvable in polynomial time?

## Acknowledgement

We would like to thank Rob Irving and the anonymous referees for detailed comments on earlier versions of this paper.

## References

[1] A. Abdulkadiroğlu and T. Sönmez. Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica*, 66(3):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 2004: the 15th Annual International Symposium on Algorithms and Computation*, volume 3341 of *Lecture Notes in Computer Science*, pages 3–15. Springer, 2004.

[3] D.J. Abraham, R.W. Irving, T. Kavitha, and K. Mehlhorn. Popular matchings. *SIAM Journal on Computing*, 37:1030–1045, 2007.

[4] D.J. Abraham and T. Kavitha. Dynamic matching markets and voting paths. In *Proceedings of SWAT 2006: the 10th Scandinavian Workshop on Algorithm Theory*, volume 4059 of *Lecture Notes in Computer Science*, pages 65–76. Springer, 2006.

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

[6] K.S. Chung. On the existence of stable roommate matchings. *Games and Economic Behavior*, 33(2):206–230, 2000.

[7] H.N. Gabow. An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems. In *Proceedings of STOC '83: the 15th Annual ACM Symposium on Theory of Computing*, pages 448–456. ACM, 1983.

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

[9] C.-C. Huang, T. Kavitha, D. Michail, and M. Nasre. Bounded unpopularity matchings. In *Proceedings of SWAT 2008: the 12th Scandinavian Workshop on Algorithm Theory*, volume 5124 of *Lecture Notes in Computer Science*, pages 127–137. Springer, 2008.

[10] R.W. Irving, T. Kavitha, K. Mehlhorn, D. Michail, and K. Paluch. Rank-maximal matchings. *ACM Transactions on Algorithms*, 2(4):602–610, 2006.

[11] T. Kavitha and M. Nasre. Optimal popular matchings. In *Proceedings of Match-UP 2008: Workshop on Matching Under Preferences – Algorithms and Complexity, held at ICALP 2008*, pages 46–54, 2008.

[12] T. Kavitha and C.D. Shah. Efficient algorithms for weighted rank-maximal matchings and related problems. In *Proceedings of ISAAC 2006: the Seventeenth International Symposium on Algorithms and Computation*, volume 4288 of *Lecture Notes in Computer Science*, pages 153–162. Springer, 2006.

[13] M. Mahdian. Random popular matchings. In *Proceedings of EC '06: the 7th ACM Conference on Electronic Commerce*, pages 238–242. ACM, 2006.

[14] D.F. Manlove and C.T.S. Sng. Popular matchings in the Capacitated House Allocation problem. In *Proceedings of ESA '06: the 14th Annual European Symposium on Algorithms*, volume 4168 of *Lecture Notes in Computer Science*, pages 492–503. Springer, 2006.

[15] R.M. McCutchen. The least-unpopularity-factor and least-unpopularity-margin criteria for matching problems with one-sided preferences. In *Proceedings of LATIN 2008: the 8th Latin-American Theoretical INformatics symposium*, volume 4957 of *Lecture Notes in Computer Science*, pages 593–604. Springer, 2008.

[16] E. McDermid and R.W. Irving. Popular matchings: Structure and algorithms. Technical Report TR-2008-292, University of Glasgow, Department of Computing Science, 2008.

[17] J. Mestre. Weighted popular matchings. In *Proceedings of ICALP '06: the 33rd International Colloquium on Automata, Languages and Programming*, volume 4051 of *Lecture Notes in Computer Science*, pages 715–726. Springer, 2006.

[18] C.T.S. Sng and D.F. Manlove. Popular matchings in the weighted capacitated house allocation problem. In *Proceedings of ACiD 2007: the 3rd Algorithms and Complexity in Durham workshop*, volume 9 of *Texts in Algorithmics*, pages 129–140. College Publications, 2007.