Parameterized Analysis of Assignment Under Multiple Preferences

Barak Steindl and Meirav Zehavi

Ben Gurion University of the Negev, Beer-Sheva, Israel

Abstract. The Assignment problem is a fundamental and well-studied problem in the intersection of Social Choice, Computational Economics and Discrete Allocation. In the Assignment problem, a group of agents expresses preferences over a set of items, and the task is to find a pareto optimal allocation of items to agents. We introduce a generalized version of this problem, where each agent is equipped with multiple incomplete preference lists: each list (called a layer) is a ranking of items in a possibly different way according to a different criterion. We introduce the concept of global optimality, which extends the notion of pareto optimality to the multi-layered setting, and we focus on the problem of deciding whether a globally optimal assignment exists. We study this problem from the perspective of Parameterized Complexity: we consider several natural parameters such as the number of layers, the number of agents, the number of items, and the maximum length of a preference list. We present a comprehensive picture of the parameterized complexity of the problem with respect to these parameters.

1 Introduction

The field of resource allocation problems has been widely studied in recent years. A fundamental and one of the most well-studied problems in this field is the Assignment problem\footnote{The problem is called Assignment in all relevant literature. Although this name is somewhat generic, to be consistent with the literature, we use it here as well.}. In the Assignment problem we are given a set of $n$ agents, and a set of $m$ items. Each agent (human, company, or any other entity) has strict preferences over a subset of items, and the objective is to allocate items to agents in an “optimal” way. Different notions of optimality have been considered in the literature, but the one that has received the most attention is pareto optimality (see, e.g., \cite{2,6,5}). Intuitively, an assignment $p$ is called pareto optimal if there is no other assignment $q$ that is at least good as $p$ for all the agents and also strictly better than $p$ for at least one agent.

Besides its theoretical interest, the problem has also practical importance. Algorithms for the Assignment problem have applications in a variety of real-world situations, such as assigning jobs to workers, campus houses to students, time stamps to users on a common machine, players to sports teams, graduating medical students to their first hospital appointments, and so on \cite{39,31,18,26}.\cite{29,13,8,27,6,5,21}
A simple and well-studied allocation mechanism is the greedy serial dictatorship mechanism, introduced by Abdulkadiroglu and Sönmez [2]. In the serial dictatorship mechanism, we draw a random permutation on the agents from the uniform distribution. The agent ordered first in the permutation is assigned to its top choice, the agent ordered second is assigned to its top choice among the remaining items, and so on. Abdulkadiroglu and Sönmez [2] proved that applying this mechanism results in a pareto optimal assignment.

In the Assignment problem, each agent has exactly one preference list. The preference lists may represent a single subjective criterion according to which each agent ranks the items. However, they may also represent a combination of different such criteria: each agent associates a score to each item per criterion, and a single preference list is derived from some weighted sum of the scores. In many cases, it is unclear how to combine scores associated with criteria of inherently incomparable nature - that is like “comparing apples with oranges”. Additionally, even if a single list can be forcefully extracted, most data is lost.

Thus, the classic model seems somewhat restrictive in real-world scenarios where people rely on highly varied aspects to rank other entities. For example, suppose that there are $n$ candidates who need to be assigned to $n$ positions. The recruiters may rank the candidates for each position according to different criteria, such as academic background, experience, impression by the interview, and so on [254]. Moreover, when assigning campus houses to students, the student may rank the houses by multiple criteria such as their location (how close the house is to their faculty), rent, size etc [35]. This motivates the employment of multiple preference lists where each preference list (called a layer) is defined by a different criterion.

In many real-world scenarios, the preferences of the agents may sometimes depend on external circumstances that may not be completely known in advance such as growth of stocks in the market, natural phenomena, outbreak of pandemics [37,35] and so on. In such cases, each layer in our generalized model can represent a possible “state” of the world, and we may seek an assignment that is optimal in as many states as possible. For instance, suppose that there is a taxi company with $n$ taxis and $m$ customers ($n > m$) that want to be picked at a specific time in future. The “cost” of each taxi depends on the time taken to reach the customer from the starting location of the taxi. Many factors (that may not be completely known a-priori) may affect the total cost such as road constructions, weather, car condition and the availability of the drivers [15,32]. The firm may suggest different possible scenarios (each represents a layer). For each scenario, the customers may be ranked differently by the taxis, and an assignment that is pareto optimal in as many layers as possible will cover most of the scenarios and will give the lowest expected total cost.

Furthermore, it is not always possible to completely take hold of preferences of some (or all) agents due to lack of information or communication, as well as security and privacy issues [30,10]. In addition, even if it is technically and
ethically feasible, it may be costly in terms of money, time, or other resources to gather all information from all the agents. In these cases, we can “complete the preferences” using different assumptions on the agents. As a result, we will have a list of preference profiles that represent different possible states of the world. An assignment that is pareto optimal in as many preference profiles as possible will be pareto optimal with high probability.

Our work is inspired by the work of Chen et al. [12], who studied the Stable Marriage problem under multiple preference lists. In contrast to the Assignment problem, the Stable Marriage problem is a two-sided matching problem, i.e. it consists of two disjoint sets of agents $A$ and $B$, such that each agent strictly ranks the agents of the opposite set (in the Assignment problem, only the agents rank the items). The objective in the Stable Marriage problem is to find a matching (called a stable matching) between $A$ and $B$ such that there do not exist agents $a \in A$ and $b \in B$ that are not matched to each other but rank each other higher than their matched partners.

Chen et al. [12] considered an extension of the Stable Marriage problem where there are $\ell$ layers of preferences, and adapted the definition of stability accordingly. Specifically, three notions of stability were defined: $\alpha$-global stability, $\alpha$-pair stability, and $\alpha$-individual stability. In their work, Chen et al. [12] studied the algorithmic complexity of finding matchings that satisfy each of these stability notions. Their notion of $\alpha$-global stability extends the original notion of stability in a natural way, by requiring the sought matching to be stable in (at least) some $\alpha$ layers. Our notion of $\alpha$-global optimality extends pareto optimality in the same way, by requiring an assignment to be pareto optimal in some $\alpha$ layers. In contrast to [12], our research focuses more on the perspective of parameterized complexity. We take into account a variety of parameters (12 focuses only on the parameter $\alpha$), and we provide parameterized algorithms, parameterized hardness results, and ETH-based lower bounds regarding them (12 focuses only on parameterized hardness results). We also study the kernelization complexity of the problem: We provide kernels, and lower bounds on kernel sizes.

Although the Assignment problem can be solved in polynomial time using a mechanism called “serial dictatorship” [2], we show that the problem becomes much harder when multiple preference lists are taken into account. So, in this paper, we study the parameterized complexity of deciding whether a globally optimal assignment exists with respect to various parameters.

Contributions and Methods. One important aspect of our contribution is conceptual: we are the first to study pareto optimality (in the Assignment problem) in the presence of multiple preference lists. This opens the door to many future studies (both theoretical and experimental) of our concept, as well as refinements or extensions thereof (see Section 6). In this work, we focus on the classical and parameterized complexity of the problem.

We consider several parameters such as the number of layers $\ell$, the number of agents $n$ (also denoted by #agents), the number of items $m$ (also denoted by

---

3 In Section 2 we further argue that the Assignment and Stable Marriage, problems, being based on different concepts of stability, are very different problems.
Parameter $k$ | Complexity Class | Running Time | Polynomial Kernel?
---|---|---|---
$\ell + d$ | para-NP-hard | $O^*(nl)^{\ell}$ | no
$\ell + (m - d)$ | para-NP-hard | $O^*(nl)^{\ell}$ | no
$n$ | FPT | $(nm)^{O(m)}$ | -
$m$ | FPT | $(nm)^{O(m)}$ | -
$m + \alpha$ | FPT | $(nm)^{O(m)}$ | -
$n + m + \alpha$ | FPT | $(nm)^{O(m)}$ | -
$n + \ell$ | FPT | $O^*((nl)^{\ell})$ | yes
$n + \ell$ | FPT | $O^*((nl)^{\ell})$ | yes

Table 1. Summary of our results for $\alpha$-Globally Optimal Assignment. Results marked with † are proved to be optimal under the exponential-time hypothesis.

#items), the maximum length of a preference list $d$, and the given number of layers $\alpha$ for which we require an assignment to be pareto optimal. The choice of these parameters is sensible because in real-life scenarios such as those mentioned earlier, some of these parameters may be substantially smaller than the input size. For instance, $\ell$, $\alpha$ and $\ell - \alpha$ are upper bounded by the number of criteria according to which the agents rank the items. Thus, they are likely to be small in practice: when ranking other entities, people usually do not consider a substantially large number of criteria (further, up until now, attention was only given to the case where $\ell = \alpha = 1$). For instance, when sports teams rank candidate players, only a few criteria such as the player’s winning history, its impact on its previous teams, and physical properties are taken into account [19]. In addition, the parameter $\ell - \alpha$ may be small particularly in cases where we want to find an assignment that is optimal with respect to as many criteria as possible. Moreover, in various cases concerning ranking of people, jobs, houses etc., people usually have a limited number of entities that they want or are allowed to ask for [14]. In these cases, the parameter $d$ is likely to be small. Moreover, in small countries (such as Israel), the number of universities, hospitals, sports teams and many other facilities and organizations is very small [13,34]. Thus, in scenarios concerning these entities, at least one among $n$ and $m$ may be small.

A summary of our results is given in Table 1.

**Fixed-Parameter Tractability and ETH Based Lower Bounds.** We prove that $\alpha$-Globally Optimal Assignment is fixed-parameter tractable (FPT) with respect to $n$ by providing an $O^*(n!)$ time algorithm that relies on the connection between pareto optimality and serial dictatorship. We then prove that the problem admits a polynomial kernel with respect to $n + \ell$ and that it is FPT with respect to $m + \ell$ by providing an exponential kernel. We also prove that the problem is slice-wise polynomial (XP) with respect to $m$ by providing an $m^{O(m)} \cdot n^{O(m)}$ time algorithm. In addition, we prove that $O^*(2^{O(t \log t)})$ is a tight lower bound on the running time (so, our $O^*(n!)$ time algorithm is essentially optimal) under ETH (defined in Section 2) for even larger parameters such as
Parameterized Analysis of Assignment Under Multiple Preferences

$t = n + m + \alpha$ and $t = n + m + (\ell - \alpha)$ using two linear parameter reductions from the $k \times k$ Permutation Clique problem. Lastly, we prove that the problem is W[1]-hard with respect to $m + \alpha$ and $m + (\ell - \alpha)$ using two parameterized reductions from Multicolored Independent Set.

**NP-Hardness.** We first prove that the problem is NP-hard for any fixed values of $\alpha$ and $\ell$ such that $2 \leq \alpha \leq \ell$ using a polynomial reduction from the Serial Dictatorship Feasibility problem that relies on a reduction by Aziz et al. [6].

We also define three polynomial-time constructions of preference profiles given an instance of 3-SAT, and we rely on them in the construction of polynomial reductions from 3-SAT to the problem, such that in the resulting instances $\ell + d$ and $\ell + (m - d)$ are bounded by fixed constants. This proves that the problem is para-NP-hard for $\ell + d$ and $\ell + (m - d)$.

**Non-Existence of Polynomial Kernels.** We prove that the problem does not admit polynomial kernels unless $\text{NP} \subseteq \text{coNP/poly}$ with respect to $n + m + \alpha$, $n + m + (\ell - \alpha)$, and $m + \ell$ using three cross-compositions (defined in Section 2) from 3-SAT that rely on the aforementioned reduction to prove para-NP-hardness.

## 2 Preliminaries

For any natural number $t$, we denote $[t] = \{1, \ldots, t\}$. We use the $O^*$ and the $\Omega^*$ notations to suppress polynomial factors in the input size, that is, $O^*(f(k)) = f(k) \cdot n^{O(1)}$ and $\Omega^*(f(k)) = \Omega(f(k)) \cdot n^{O(1)}$.

**The Assignment problem.** An instance of the Assignment problem is a triple $(A, I, P)$ where $A$ is a set of $n$ agents $\{a_1, \ldots, a_n\}$, $I$ is a set of $m$ items $\{b_1, \ldots, b_m\}$, and $P = (\prec_{a_1}, \ldots, \prec_{a_n})$, called the preference profile, contains the preferences of the agents over the items, where each $\prec_{a_i}$ encodes the preferences of $a_i$ and is a linear order over a subset of $I$ (preferences are allowed to be incomplete). We refer to such linear orders as preference lists. If $b_j \prec_{a_i} b_r$, we say that agent $a_i$ prefers item $b_r$ over item $b_j$, and we write $b_j \leq_{a_i} b_r$ if $b_j \prec_{a_i} b_r$ or $b_j = b_r$. Item $b$ is acceptable by agent $a$ if $b$ appears in $a$’s preference list. An assignment is an allocation of items to agents such that each agent is allocated at most one item, and each item is allocated to at most one agent. Since the preferences of the agents may be incomplete, or the number of items may be smaller than the number of agents, some agents may not have available items to be assigned to. In order to deal with this case, we define a special item $b_0$, seen as the least preferred item of each agent, and will be used as a sign that an agent is not allocated an item. Throughout this paper, we assume that $b_0$ is not part of the input item set, and that it appears at the end of every preference list (we will not write $b_0$ explicitly in the preference lists). We formally define assignments as follows:

**Definition 1.** Let $A = \{a_1, \ldots, a_n\}$ be a set of $n$ agents and let $I = \{b_1, \ldots, b_m\}$ be a set of $m$ items. A mapping $p : A \rightarrow I \cup \{b_0\}$ is called an assignment if for each $i \in [n]$, it satisfies one of the following conditions:
1. \( p(a_i) = b_0 \).
2. Both \( p(a_i) \in I \) and for each \( j \in [n] \setminus \{i\} \), \( p(a_i) \neq p(a_j) \).

We refer to \( p \) as legal if, for each \( i \in [n] \), it satisfies \( p(a_i) = b_0 \) or \( p(a_i) \in I \) is acceptable by \( a_i \). For brevity, we will omit the term “legal”, referring to a legal assignment just as an assignment.

Moreover, when we write a set in a preference list, we assume that its elements are ordered arbitrarily, unless stated otherwise. In the Assignment problem, we are given such triple \((A, I, P)\), and we seek a pareto optimal assignmen.

**Pareto Optimality.** There are different ways to define optimality of assignments, but the one that received the most attention in the literature is **pareto optimality.** Informally speaking, an assignment \( p \) is **pareto optimal** if there does not exist another assignment \( q \) that is “at least as good” as \( p \) for all the agents, and is “better” for at least one agent. It is formally defined as follows.

**Definition 2.** Let \( A = \{a_1, \ldots, a_n\} \) be a set of agents, and let \( I \) be a set of items. An assignment \( p : A \to I \cup \{b_0\} \) is pareto optimal if there does not exist another assignment \( q : A \to I \cup \{b_0\} \) that satisfies:

1. \( p(a_i) \leq_\alpha q(a_i) \) for every \( i \in [n] \).
2. There exists \( i \in [n] \) such that \( p(a_i) <_\alpha q(a_i) \).

If such an assignment \( q \) exists, we say that \( q \) pareto dominates \( p \).

**The Assignment problem and the Stable Marriage problem.** On first sight, it may seem that the Assignment problem is a special case of the Stable Marriage problem with indifferences (ties) [24] (the preferences of the agents remain the same, and all the items rank all the agents equally). We stress that, in fact, it is a very different problem since pareto optimality is not equivalent to the stability notions defined by Irving [24] for the Stable Marriage problem with indifference. Consider the following instance of the Assignment problem with three agents \( a_1, a_2, a_3 \), and three items \( b_1, b_2, b_3 \):

- \( a_1 : b_1 > b_2 > b_3 \)
- \( a_2 : b_3 > b_2 > b_1 \)
- \( a_3 : b_3 > b_1 > b_2 \)

We transform this instance to an instance of Stable Marriage with indifference by making all the items to rank \( a_1, a_2 \) and \( a_3 \) equally as follows:

---

4 All the “optimal” assignments that we construct in this paper will be legal in a sufficient number of layers, where they are claimed to be pareto optimal.
In the context of Stable Marriage with indifference, a matching \( M \) is called weakly stable if there is no pair of agents each of whom strictly prefers the other to its matched partner. In addition, \( M \) is called super stable if there is no pair of agents each of whom either strictly prefers the other to its partner or is indifferent between them. Moreover, \( M \) is called strongly stable if there is no pair of agents such that the first one strictly prefers the second to its partner, and the second one strictly prefers the first to its partner or is indifferent between them (see [24] for the formal definitions).

Consider the matching \( p \) that satisfies \( p(a_i) = b_i \) for every \( i \in \{1, 2, 3\} \). Then, \( p \) is pareto optimal for the Assignment instance, but it is not strongly stable for the constructed Stable Marriage instance since \( \{a_2, b_3\} \) is a blocking pair. Suppose the matching \( q \) that satisfies \( q(a_1) = b_2, q(a_2) = b_1 \) and \( q(a_3) = b_3 \). Then, \( q \) is not pareto optimal for the Assignment instance because it admits the trading cycle \((a_1, b_2, a_2, b_1)\), but it is weakly stable and even super stable for the Stable Marriage instance. Thus, we conclude that the problems are different from each other.

We first give some well-known characterizations of assignments, and after that we introduce new concepts of optimality and three new multi-layered assignment problems.

Intuitively, an assignment admits a trading cycle if there exists a set of agents who all benefit by exchanging their allocated items among themselves. For example, a simple trading cycle among two agents \( a \) and \( b \) occurs when agent \( a \) prefers agent \( b \)'s item over its own item, and agent \( b \) prefers agent \( a \)'s item over its own item. Both \( a \) and \( b \) would benefit from exchanging their items. Formally, a trading cycle is defined as follows.

**Definition 3.** An assignment \( p \) admits a trading cycle \((a_{i_0}, b_{j_0}, a_{i_1}, b_{j_1}, \ldots, a_{i_{k-1}}, b_{j_{k-1}})\) if for each \( r \in \{0, \ldots, k - 1\} \), we have that \( p(a_{i_r}) = b_{j_r} \) and \( b_{j_r} <_{a_{i_r}} b_{j_r+1} \) (mod \( k \)).

**Definition 4.** An assignment \( p \) admits a self loop if there exist an agent \( a_i \) and an item \( b_j \) such that \( b_j \) is not allocated to any agent by \( p \), and \( p(a_i) <_{a_i} b_j \).

**Proposition 1 (Folklore; see, e.g., Aziz et al. [6, 5]).** An assignment \( p \) is pareto optimal if and only if it does not admit trading cycles and self loops.

By this proposition, the problem of checking whether an assignment admits trading cycles or self loops can be reduced to the problem of checking whether the directed graph defined next contains cycles. For an instance \((A, I, P)\) and an assignment \( p \), the corresponding trading graph is the directed graph defined as follows. Its vertex set is \( A \cup I \), and there are three types of edges:
For each $a \in A$ such that $p(a) \neq b_\emptyset$, there is a directed edge from $p(a)$ to $a$. Namely, each allocated item points to its owner.

For each agent $a \in A$, there is an edge from $a$ to all the items it prefers over its assigned item $p(a)$ (if $p(a) = b_\emptyset$, $a$ points to all its acceptable items).

Each item with no owner points to all the agents that accept it.

**Proposition 2 (Folklore; see, e.g., Aziz et al. [6,5]).** An assignment $p$ is pareto optimal if and only if its corresponding trading graph does not contain cycles.

**Example.** Suppose that $A = \{a_1, a_2, a_3, a_4, a_5\}$ and $I = \{b_1, b_2, b_3, b_4, b_5\}$. Assume that the preferences of the agents are defined as follows.

- $a_1 : b_4 > b_1 > b_2 > b_5$
- $a_2 : b_1 > b_4 > b_5$
- $a_3 : b_2 > b_1$
- $a_4 : b_3 > b_5$
- $a_5 : b_5$

Let $p : A \rightarrow I \cup \{b_\emptyset\}$ be an assignment such that $p(a_1) = b_2$, $p(a_2) = b_4$, $p(a_3) = b_1$, $p(a_4) = b_5$, and $p(a_5) = b_\emptyset$. The trading graph of the preference profile with respect to $p$ is:

![Trading Graph](image)

Observe that agents $a_1$, $a_3$ and $a_5$ admit the trading cycle $(a_1, b_2, a_2, b_4, a_3, b_1)$, and that agent $a_4$ admits a self loop with $b_3$. By Proposition 2, $p$ is not pareto optimal. If $a_1$, $a_2$, and $a_3$ exchange their items, $a_4$ gets $b_3$, and $a_5$ gets $b_2$, we have a pareto optimal assignment $q$ in which $q(a_1) = b_4$, $q(a_2) = b_1$, $q(a_3) = b_2$, $q(a_4) = b_3$ and $q(a_5) = b_5$.

A simple assignment mechanism is the greedy *serial dictatorship* mechanism. For a given permutation over the agents, the mechanism takes agents in turns, one in each turn, according to the permutation. That is, the agent which is ordered first allocates its most preferred item, the second allocates its most preferred item among the remaining items, and so on. If at some point, an agent has no available item to allocate in its preference list, it allocates $b_\emptyset$. We say
that an assignment $p$ is a possible outcome of serial dictatorship if there exists a permutation $\pi$ such that applying serial dictatorship with respect to $\pi$ results in $p$.

**Proposition 3 (Abdulkadiroglu and Sönmez [2]).** An assignment $p$ is pareto optimal if and only if it is a possible outcome of serial dictatorship.

**Corollary 1.** A pareto optimal assignment always exists and can be found in polynomial time, and the number of pareto optimal assignments for an instance with $n$ agents is at most $n!$.

Proposition 3 yields a surjective mapping from the set of permutations on the agents to the set of pareto optimal assignments. This implies an upper bound of $n!$ on the number of pareto optimal assignments. Observe that this bound is tight: consider an instance where there is an equal number of agents and items, and all the agents share the same complete preference list. Observe that each permutation gives us a unique assignment after applying serial dictatorship with respect to it. Thus, there exist exactly $n!$ pareto optimal assignments.

**Generalization of the Assignment Problem.** We introduce a generalized version of the Assignment problem where there are $\ell$ layers of preferences. For each $j \in [\ell]$, we refer to $<_{a_i}^{(j)}$ as $a_i$’s preference list in layer $j$. The preference profile in layer $j$ is the collection of all the agents’ preference lists in layer $j$, namely, $P_j = (<_a^{(j)},...,<_a^{(j)})$. We say that assignment $p$ is pareto optimal in layer $j$ if it is pareto optimal in the single-layered instance $(A, I, P_j)$. To adapt the notion of optimality to the new context, we introduce a natural generalization requiring an assignment to be optimal in a given number of layers.

**Definition 5.** An assignment $p$ is $\alpha$-globally optimal for an instance $(A, I, P_1, \ldots, P_\ell, \alpha)$ if there exist $\alpha$ layers $i_1, \ldots, i_\alpha \in [\ell]$ such that $p$ is pareto optimal in layer $i_j$ for each $j \in [\alpha]$.

Thus, the new problem is defined as follows.

\[\alpha\text{-Globally Optimal Assignment}\]

**Input:** $(A, I, P_1, \ldots, P_\ell, \alpha)$, where $A$ is a set of $n$ agents, $I$ is a set of $m$ items, $P_i$ is the preference profile in layer $i$ for each $i \in [\ell]$, and $\alpha \in [\ell]$.

**Question:** Does an $\alpha$-globally optimal assignment exist?

**Example.** Consider the following instance, where the agent set is $A = \{a_1, a_2, a_3, a_4\}$, the item set is $I = \{b_1, b_2, b_3, b_4\}$, and there are four layers, defined as follows.

Layer 1:

- $a_1 : b_1$
- $a_2 : b_3 > b_2 > b_1$
- $a_3 : b_3 > b_1$
- $a_4 : b_2 > b_1 > b_3$
Fig. 1. The trading graph with respect to $p$ in the fourth layer, where $p$ admits the trading cycle $(a_1, b_1, a_3, b_3, a_2, b_2)$.

Layer 2:

- $a_1$: $b_2 > b_1$
- $a_2$: $b_2 > b_3$
- $a_3$: $b_1 > b_2 > b_3$
- $a_4$: $b_3$

Layer 3:

- $a_1$: $b_2 > b_1$
- $a_2$: $b_4 > b_2 > b_1$
- $a_3$: $b_1 > b_3$
- $a_4$: $b_2 > b_1 > b_3$

Layer 4:

- $a_1$: $b_3 > b_1 > b_2$
- $a_2$: $b_1 > b_2$
- $a_3$: $b_2 > b_3$
- $a_4$: $\emptyset$

Consider an assignment $p$ in which $a_i$ gets $b_i$ for every $i \in \{1, 2, 3\}$, and $a_4$ gets $b_\emptyset$. The assignment is 2-globally optimal since it is pareto optimal in the first two layers. To see this, apply serial dictatorship in the first layer with respect
to the permutation \((a_1, a_3, a_2, a_4)\), and in the second layer with respect to the permutation \((a_2, a_1, a_3, a_4)\), and verify that \(P\) is the outcome of both runs. In contrast, \(P\) is not pareto optimal in the third layer since it admits a self loop among \(a_2\) and \(b_4\) (\(b_4\) is available and is preferred by \(a_2\) over its assigned item \(b_2\)). Furthermore, assignment \(P\) is not pareto optimal in the fourth layer because it admits a trading cycle \((a_1, b_1, a_3, b_3, a_2, b_2)\) (see Figure 1): if \(a_1, a_2\) and \(a_3\) trade their items, we get a new assignment \(Q\) in which \(a_1\) gets \(b_3\), \(a_2\) gets \(b_1\), and \(a_3\) gets \(b_2\); assignment \(Q\) pareto dominates \(P\) in the fourth layer, and we can verify that it is also pareto optimal in this layer.

**Parameterized Complexity.** Let \(\Pi\) be an NP-hard problem. In the framework of Parameterized Complexity, each instance of \(\Pi\) is associated with a parameter \(k\). Here, the goal is to confine the combinatorial explosion in the running time of an algorithm for \(\Pi\) to depend only on \(k\). Formally, we say that \(\Pi\) is fixed-parameter tractable (FPT) if any instance \((I, k)\) of \(\Pi\) is solvable in time \(f(k) \cdot |I|^{O(1)}\), where \(f\) is an arbitrary computable function of \(k\). A weaker request is that for every fixed \(k\), the problem \(\Pi\) would be solvable in polynomial time. Formally, we say that \(\Pi\) is slice-wise polynomial (XP) if any instance \((I, k)\) of \(\Pi\) is solvable in time \(f(k) \cdot |I|^{g(k)}\), where \(f\) and \(g\) are arbitrary computable functions of \(k\). Nowadays, Parameterized Complexity supplies a rich toolkit to design FPT and XP algorithms.

Parameterized Complexity also provides methods to show that a problem is unlikely to be FPT. The main technique is the one of parameterized reductions analogous to those employed in classical complexity. Here, the concept of \(W[1]\)-hardness replaces the one of \(NP\)-hardness, and for reductions we need not only construct an equivalent instance in FPT time, but also ensure that the size of the parameter in the new instance depends only on the size of the parameter in the original one.

**Definition 6. (Parameterized Reduction)** Let \(\Pi\) and \(\Pi'\) be two parameterized problems. A parameterized reduction from \(\Pi\) to \(\Pi'\) is an algorithm that, given an instance \((I, k)\) of \(\Pi\), outputs an instance \((I', k')\) of \(\Pi'\) such that:

- \((I, k)\) is a Yes-instance of \(\Pi\) if and only if \((I', k')\) is a Yes-instance of \(\Pi'\).
- \(k' \leq g(k)\) for some computable function \(g\).
- The running time is \(f(k) \cdot |I|^{O(1)}\) for some computable function \(f\).

If there exists such a reduction transforming a problem known to be \(W[1]\)-hard to another problem \(\Pi\), then the problem \(\Pi\) is \(W[1]\)-hard as well. Central \(W[1]\)-hard-problems include, for example, deciding whether a nondeterministic single-tape Turing machine accepts within \(k\) steps, CLIQUE parameterized by solution size, and INDEPENDENT SET parameterized by solution size. To show that a problem \(\Pi\) is not XP unless \(P=NP\), it is sufficient to show that there exists a fixed \(k\) such \(\Pi\) is \(NP\)-hard. Then, the problem is said to be para-NP-hard.

A companion notion to that of fixed-parameter tractability is the one of a polynomial kernel. Formally, a parameterized problem \(\Pi\) is said to admit a polynomial compression if there exists a (not necessarily parameterized) problem \(\Pi'\) and a polynomial-time algorithm that given an instance \((I, k)\) of \(\Pi\), outputs
an equivalent instance $I'$ of $II'$ (that is, $(I, k)$ is a Yes-instance of $II$ if and only if $I'$ is a Yes-instance of $II'$) such that $|I'| \leq p(k)$ where $p$ is some polynomial that depends only on $k$. In case $II' = II$, we further say that $II$ admits a polynomial kernel. For more information on Parameterized Complexity, we refer the reader to recent books such as [17,16,20].

**Non-Existence of a Polynomial Compression.** Our proof of the “unlikely existence” of a polynomial kernel for $\alpha$-Globally Optimal Assignment relies on the well-known notion of cross-composition, defined as follows.

**Definition 7 (Cross-Composition).** A (not parameterized) problem $II$ cross-composes into a parameterized problem $II'$ if there exists a polynomial-time algorithm, called a cross-composition, that given instances $I_1, I_2, \ldots, I_t$ of $II$ for some $t \in \mathbb{N}$ that are of the same size $s$ for some $s \in \mathbb{N}$, outputs an instance $(I, k)$ of $II'$ such that the following conditions are satisfied.

- $k \leq p(s + \log t)$ for some polynomial $p$.
- $(I, k)$ is a Yes-instance of $II'$ if and only if at least one of the instances $I_1, I_2, \ldots, I_t$ is a Yes-instance of $II$.

**Proposition 4 ([7,8]).** Let $II$ be an NP-hard (not parameterized) problem that cross-composes into a parameterized problem $II'$. Then, $II'$ does not admit a polynomial compression, unless $NP \subseteq coNP/poly$.

To obtain (essentially) tight conditional lower bounds for the running times of algorithms, we rely on the well-known Exponential-Time Hypothesis (ETH) [22,23,11]. To formalize the statement of ETH, first recall that given a formula $\varphi$ in conjunctive normal form (CNF) with $n$ variables and $m$ clauses, the task of CNF-SAT is to decide whether there is a truth assignment to the variables that satisfies $\varphi$. In the $p$-CNF-SAT problem, each clause is restricted to have at most $p$ literals. ETH asserts that 3-CNF-SAT cannot be solved in time $O(2^{o(n)})$.

### 3 Fixed-Parameter Tractability and ETH Based Lower Bounds

We first prove that $\alpha$-Globally Optimal Assignment is FPT with respect to the parameter $n = \#agents$ by presenting an algorithm with running time $O^*(n!)$. Afterwards, we prove that this is essentially the best possible running time for this parameter (and even for a larger parameter) under ETH. Let us first prove the following lemma.

**Lemma 1.** Let $(A, I, P_1, \ldots, P_\ell, \alpha)$ be an instance of $\alpha$-Globally Optimal Assignment, and let $p$ be an assignment. Let $G_i$ denote the trading graph of $P_i$ with respect to $p$ for each $i \in [\ell]$. Then $p$ is $\alpha$-globally optimal for the instance if and only if there exist $\alpha$ trading graphs among $G_1, \ldots, G_\ell$ that contain no cycles.
Parameterized Analysis of Assignment Under Multiple Preferences

**Algorithm 1:** Algorithm for α-Globally Optimal Assignment.

```
input : An instance \((A, I, P_1, \ldots, P_\ell, \alpha)\)
output: Does an \(\alpha\)-globally optimal assignment exist?

foreach \(i \in [\ell]\) do
  foreach permutation \(\pi\) on \(A\) do
    apply serial dictatorship on profile \(P_i\) with respect to \(\pi\), to obtain
    an assignment \(p\)
    \(\text{count} \leftarrow 1\)
    foreach \(j \in [\ell] \setminus \{i\}\) do
      \(G \leftarrow P_j\)'s trading graph with respect to \(p\)
      if \(G\) contains no cycles then
        \(\text{count} \leftarrow \text{count} + 1\)
      end
    end
    if \(\text{count} \geq \alpha\) then
      return Yes
    end
  end
end
return No
```

**Proof.** \((\Rightarrow)\) Assume that \(p\) is \(\alpha\)-globally optimal. Then there exist \(\alpha\) layers \(i_1, \ldots, i_\alpha\) such that for each \(j \in [\alpha]\), \(p\) is pareto optimal in \(P_{i_j}\). By Proposition 2 for each \(j \in [\alpha]\), \(G_{i_j}\) does not contain cycles.

\((\Leftarrow)\) Assume that there exist \(\alpha\) layers \(i_1, \ldots, i_\alpha\), such that for each \(j \in [\alpha]\), \(G_{i_j}\) does not contain cycles. By Proposition 2 \(p\) is pareto optimal in each \(P_{i_j}\), implying that it is \(\alpha\)-globally optimal. \(\square\)

**Theorem 1.** There exists an \(O^*(n!)\) algorithm for \(\alpha\)-Globally Optimal Assignment.

**Proof.** We present a brute-force algorithm (described formally in Algorithm 1). The algorithm enumerates all possible pareto optimal assignments for each layer, using serial dictatorship. For each assignment, it constructs the corresponding trading graphs, and checks whether there exist \(\alpha\) graphs with no cycles.

The running time of the algorithm is \(O^*(n!)\), since it iterates over \(\ell n!\) assignments, and for each assignment, it takes polynomial time to construct \(\ell - 1\) trading graphs, and to count how many of them contain no cycles.

Let us now prove that the algorithm returns \(\text{Yes}\) if and only if the input is a \(\text{Yes}\)-instance.

\((\Rightarrow)\) Suppose that the algorithm returns \(\text{Yes}\). This implies that there exist a layer \(i \in [\ell]\) and a permutation \(\pi\) on \(A\), such that serial dictatorship on \(P_i\) with respect to \(\pi\) produces an assignment \(p\) satisfying that there exist \(\alpha - 1\) graphs among the trading graphs of \(\{P_j\mid j \in [\ell], j \neq i\}\) with respect to \(p\) which do not
contain cycles. By Proposition 3, \( p \) is pareto optimal in \( P_i \), and by Lemma 1, it is pareto optimal in \( \alpha - 1 \) preference profiles among \( \{ P_j \mid j \in [\ell], j \neq i \} \). This implies that \( p \) is \( \alpha \)-globally optimal.

\((\Leftarrow)\) Suppose we are dealing with a Yes-instance; then, there exists an assignment \( p \), and \( \alpha \) layers \( i_1, \ldots, i_\alpha \) in which \( p \) is pareto optimal. Proposition 3 implies that for each of these layers \( i_j \), there exists a permutation \( \pi_j \) such that applying serial dictatorship on \( P_{i_j} \) with respect to \( \pi_j \) results in \( p \). Thus, when the algorithm reaches one of the layers \( i_1, \ldots, i_\alpha \) and its corresponding permutation for the first time, it generates \( p \). By Lemma 1, the algorithm verifies correctly that it is \( \alpha \)-globally optimal.

We first give a simple lemma that will help us to design a polynomial kernel for \( \alpha \)-Globally Optimal Assignment with respect to the parameter \( n + \ell \).

**Lemma 2.** Let \((A, I, P)\) be an instance of the Assignment problem where \( |A| = n \). Then, for any agent \( a \in A \) and pareto optimal assignment, \( a \) is assigned to \( b_0 \) or to one of the \( n \) most preferred items in its preference list.

**Proof.** Let \( p \) be a pareto optimal assignment for \((A, I, P)\). By Proposition 3, there exists a permutation \( \pi = (a_1, \ldots, a_n) \) on \( A \) such that applying serial dictatorship with respect to \( \pi \) results in \( p \). Let \( i \in [n] \). When the mechanism is in the \( i \)-th step, it has already allocated at most \( i - 1 \) items from \( I \) to agents \( a_1, \ldots, a_{i-1} \). Hence, \( p(a_i) = b_0 \), or \( p(a_i) \) is the most preferred item of \( a_i \) among the remaining items in its preference list, and is ranked at position \( j \leq i \leq n \).

**Theorem 2.** \( \alpha \)-Globally Optimal Assignment admits a kernel of size \( \mathcal{O}(n^2) \).

**Proof.** Given an instance of \( \alpha \)-Globally Optimal Assignment \( I_1 = (A, I, P_1, \ldots, P_\ell, \alpha) \), the kernel reduces each preference profile \( P_i \) to a preference profile \( P'_i \) by keeping only the \( n \) first-ranked items in each preference list.
Corollary 2. Let \( I' \) be a set containing the items ranked in the first \( n \) positions in some preference list in \( I_1 \); then, \( |I'| \leq \ell n^2 \). The resulting instance is \( I_2 = (A, I', P_1', \ldots, P_{\ell}' , \alpha) \), and it satisfies \( |I_2| = O(\ell n^2) \). We claim that an assignment \( p \) is \( \alpha \)-globally optimal for the instance \( I_1 \) if and only if it is \( \alpha \)-globally optimal for the instance \( I_2 \).

\((\Rightarrow)\) Let \( p \) be an \( \alpha \)-globally optimal assignment for \( I_1 \); then, there exist \( \alpha \) layers \( i_1, \ldots, i_\alpha \), such that for each \( j \in [\alpha] \), \( p \) is pareto optimal in \( P_{i_j} \). By Lemma 2, each agent \( a \in A \) is assigned an item that appears in the first \( n \) items in its preference list in each \( P_{i_j} \). Hence, \( p \) assigns an acceptable item to \( a \) in each \( P_{i_j}' \). Moreover, it is pareto optimal in each \( P_{i_j}' \), as otherwise, it would contradict \( p \)'s optimality in each \( P_{i_j} \).

\((\Leftarrow)\) Let \( p \) be an \( \alpha \)-globally optimal assignment for \( I_2 \). Then by Proposition 3, there exist \( \alpha \) profile-permutation pairs \((P_{i_1}', \pi_1), \ldots, (P_{i_\alpha}', \pi_\alpha)\), such that for each \( j \in [\alpha] \), applying serial dictatorship on profile \( P_{i_j}' \) with respect to \( \pi_j \) results in \( p \). Observe that applying serial dictatorship on each \( P_{i_j} \) with respect to the same permutation \( \pi_j \) results also in \( p \). Therefore, by Proposition 3, \( p \) is also \( \alpha \)-globally optimal for \( I_1 \).

Equality 2. \( \alpha \)-GLOBALLY OPTIMAL ASSIGNMENT admits a polynomial kernel with respect to the parameter \( k = n + \ell \).

Before we present an exponential kernel for \( \alpha \)-GLOBALLY OPTIMAL ASSIGNMENT with respect to the parameter \( k = m + \ell \), let us define the following.

Definition 8. Let \( Q = (A, I, P_1, \ldots, P_\ell, \alpha) \) be an instance of \( \alpha \)-GLOBALLY OPTIMAL ASSIGNMENT and \( u \in A \). The agent class of \( u \) in \( Q \), \( C(u, Q) \), is the tuple that contains the preference lists of \( u \) in all the layers, namely, \( C(u, Q) = (\langle u_{1,1}, \ldots, u_{1,\ell} \rangle, \ldots, \langle u_{n,1}, \ldots, u_{n,\ell} \rangle) \). Define \( D = \{ B \subseteq I \times I \mid B \text{ is a linear order}\} \). For a given tuple of length \( \ell \) consisting of linear orderings on subsets of \( I \), \( C \subseteq D^\ell \), define \( A(C, Q) = \{ a \in A \mid C(a, Q) = C \} \).

Theorem 3. \( \alpha \)-GLOBALLY OPTIMAL ASSIGNMENT admits a kernel of size \( O((m!)^{\ell+1}) \). Thus, it is FPT with respect to \( m + \ell \).

Proof. Given an instance of \( \alpha \)-GLOBALLY OPTIMAL ASSIGNMENT \( Q = (A, I, P_1, \ldots, P_\ell, \alpha) \), the kernelization algorithm works as follows (formally described in Algorithm 2). It removes from \( A \) agents which share the same agent class together with all their preference lists, such that in the resulting instance there will be at most \( m+1 \) agents in the set \( A(C(a, Q), Q) \), for each \( a \in A \). Intuitively, the idea behind the correctness is that since there are \( m \) items, at most \( m \) agents in \( A(C(a, Q), Q) \) will be assigned to items; we keep at most \( m+1 \) agents (rather than \( m \)) in each agent class to cover the case where an agent is assigned to \( b \) and admits a self-loop. The kernelization algorithm clearly runs in a polynomial time.

Assume that we run the kernel on \( I_1 = (A_1, I, P_1, \ldots, P_\ell, \alpha) \) to obtain an instance \( I_2 = (A_2, I, Q_1, \ldots, Q_\ell, \alpha) \). We first observe the following:

Claim 1. \( |I_2| = O((m!)^{\ell+1}) \).
Proof. Note that there exist $\ell$ different combinations of such $I$ an assignment $q$ from $A$ such that for each $a_1, a_2 \in A$, $|A(I, q, a_1, a_2)| = m+1$ then there exist $|A(I, q, a_1, a_2)| - m - 1$ arbitrary agents from $A$ in these sets are allocated the same number of items by $A$. Thus, $|A(I, q, a_1, a_2)| = m+1$. Then, there exist $\alpha$ items which are allocated by $A$ to agents in $A$ such that admits a self loop in $A$. For the sake of contradiction, suppose there exists an assignment $q : A_2 \rightarrow I \cup \{b_0\}$ for the reduced instance as follows: For each $a \in A_2$, let $p(A_1(I(a, I_1), I_1))$ denote the set of items allocated to the agents $A_1(I(a, I_1), I_1)$ by $p$. We allocate the items in $p(A_1(I(a, I_1), I_1))$ to agents in $A_2(I, I_2)$. Agents that do not have available items are assigned to $b_0$. First, observe that $q$ allocates all the items which are allocated by $p$ since there are at most $m$ items, and the algorithm keeps all or exactly $m + 1$ agents from each set $A_1(I(a, I_1), I_1)$. As a result, $q$ cannot admit self loops in layers $i_1, \ldots, i_\alpha$ of $I_2$. Formally, the sets $A_1(I(a, I_1), I_1)$ and $A_2(I, I_2)$ satisfy $|A_2(I, I_2)| \leq |A_1(I, I_1)|$. Since the agents in these sets are allocated the same number of items by $p$ and $q$, if there exists an agent in $A_2(I, I_2)$ that admits a self loop in $I_2$, there must exist an agent in $A_1(I, I_1)$ that admits a self loop in $I_1$. Second, we claim that $q$ does not admit trading cycles in these layers. For the sake of contradiction, suppose there exists a layer $i_j$ in $I_2$, and $t$ agents $a_1, \ldots, a_t \in A_2$ that admit a trading cycle $(a_1, q(a_1), \ldots, a_t, q(a_t))$ in $Q_{i_j}$. By the construction of $q$, notice that there exist $t$ agents $a_1, \ldots, a_t \in A_1$, such that for each $i \in [t]$, $C(a_i, I_1) = C(a_i, I_2)$, and $q(a_i) = p(a_i)$. Then, $p$ admits the trading cycle $(a_1, p(a_1), \ldots, a_t, p(a_t))$ in $P_{i_j}$. This gives a contradiction to Proposition \[11\].

$(\Leftarrow)$: Assume that there exists an $\alpha$-globally optimal assignment $q$ for $I_2$. Then there exist $\alpha$ layers $i_1, \ldots, i_\alpha$ in $I_2$ in which $q$ is pareto optimal. We denote
an assignment \( p \) for \( I_1 \) by \( p(a) = \begin{cases} q(a) & a \in A_2 \\ b_0 & \text{otherwise} \end{cases} \), and we claim that \( p \) is pareto optimal in layers \( i_1, \ldots, i_\alpha \) in \( I_1 \). By the construction of \( p \), for each \( a_1 \in A_1 \setminus A_2 \), there exists an agent \( a_2 \in A_2 \) such that \( C(a_1, I_1) = C(a_2, I_2) \) and \( p(a_1) = q(a_2) \). Namely, there exists a mapping \( f \) from agents in \( A_1 \) to agents in \( A_2 \) such that for each \( a_1 \in A_1 \), \( C(a_1, I_1) = C(f(a_1), I_2) \) and \( p(a_1) = q(f(a_1)) \). If \( p \) admits a trading cycle \((a_1, p(a_1), \ldots, a_r, p(a_r))\) in some layer \( i_j \) of \( I_1 \), then \( q \) admits the trading cycle \((f(a_1), q(f(a_1)), \ldots, f(a_r), q(f(a_r)))\) in layer \( i_j \) of \( I_2 \). If \( p \) admits a self loop in layer \( i_j \) of \( I_1 \) with agent \( a_1 \in A_1 \), then \( q \) admits a self loop with agent \( f(a_1) \) in layer \( i_j \) of \( I_2 \). Thus by Proposition 1, we conclude that \( p \) is \( \alpha \)-globally optimal in \( I_1 \).

**Corollary 3.** \( \alpha \)-Globally Optimal Assignment is FPT with respect to the parameter \( k = m + \ell \).

By Theorems 2 and 3, we conclude the following.

**Corollary 4.** \( \alpha \)-Globally Optimal Assignment is solvable in time \( O^*((m!)^{\ell+1}) \).

**Theorem 4.** \( \alpha \)-Globally Optimal Assignment is solvable in time \( (nm)^{O(m)} \).

**Proof.** We present a simple brute-force algorithm. The algorithm simply iterates over all subsets of items \( I' \subseteq I \). For each subset, it iterates over all subsets \( A' \subseteq A \) such that \( |A'| = |I'| \). For each \( a \notin A' \), the algorithm allocates \( b_0 \), and it tries all possible \( |I'|! \) different ways to allocate the items in \( I' \) to the agents in \( A' \) (it skips allocations that allocate items that are not acceptable by their owners in more than \( \ell - \alpha + 1 \) layers). The algorithm constructs the corresponding trading graphs, and verifies in polynomial time whether the current assignment is \( \alpha \)-globally optimal. Hence, the running time of the algorithm is \( \sum_{t=0}^{m} \binom{n}{t} \cdot \binom{\ell}{t} \cdot t! \cdot (n+m)^{O(1)} \leq m \cdot 2^m \cdot n^\ell \cdot m! \cdot (n+m)^{O(1)} = (nm)^{O(m)} \). \( \square \)

Before we continue with our next results, let us discuss a simple property that will help in many of our proofs.

**Definition 9.** Let \((A, I, P)\) be an instance of the Assignment problem and suppose that \( P = \{ <a | a \in A \} \). We say that agents \( a_1, a_2 \in A \) respect each other if there exists a linear order on a subset of \( I \), \( \ll \subseteq I \times I \), such that both \( <a_1 \ll \ll \) and \( <a_2 \ll \ll \).

**Lemma 3.** Let \((A, I, P)\) be an instance of the Assignment problem, such that there exist agents \( a_1, \ldots, a_r \in A \) where for each \( i, j \in [r], a_i \) and \( a_j \) respect each other. Then, for every assignment \( p : A \to I \cup \{b_0\} \), \( p \) does not admit a trading cycle among the agents \( a_1, \ldots, a_r \).

**Proof.** Towards a contradiction, suppose there exist an assignment \( p \) which admits a trading cycle \((a_1, p(a_1), \ldots, a_r, p(a_r))\) (notice that \( r \geq 2 \)). Since \( a_1, \ldots, a_r \)
pairwise respect each other, there exists a linear order \( \preceq \subseteq I \times I \), such that for each \( i \in [r] \), \( a_i \preceq \). This implies that \( p(a_1) \preceq p(a_2) \preceq \ldots \preceq p(a_r) \). Since \( p(a_1) \prec a_i \prec p(a_1) \), we have that \( p(a_r) \prec p(a_1) \), a contradiction to \( \prec \) being a linear order.

We now prove that \( \mathcal{O}^*(k!) \) is a (tight) lower bound on the running time of any algorithm for \( \alpha \)-GLOBALLY OPTIMAL ASSIGNMENT under the \( \text{ETH} \), even for larger parameters than \( n \) such as \( k = n + m + \alpha \) and \( k = n + m + (\ell - \alpha) \). So, the algorithm in Theorem 1 is optimal (in terms of running time).

**Proposition 5** (Cygan et al. [16]). Suppose that there is a polynomial-time parameterized reduction from problem \( A \) to problem \( B \) such that if the parameter of an instance of \( A \) is \( k \), then the parameter of the constructed instance of \( B \) is at most \( g(k) \) for some nondecreasing function \( g \). Then an \( \mathcal{O}^*\left(2^{o(f(k))}\right) \)-time algorithm for \( B \) for some nondecreasing function \( f \) implies an \( \mathcal{O}^*(2^{o(f(g(k)))}) \)-time algorithm for \( A \).

**Theorem 5.** Unless \( \text{ETH} \) fails, there does not exist an algorithm for \( \alpha \)-GLOBALLY OPTIMAL ASSIGNMENT with running time \( \mathcal{O}^*\left(2^{o(k \log k)}\right) \) where \( k = n + m + \alpha \).

**Proof.** We provide a linear parameter reduction from \( k \times k \) Permutation Clique. In the \( k \times k \) Permutation Clique problem, we are given a graph \( G \) where the vertices are elements of a \( k \times k \) table, namely, \( V(G) = [k] \times [k] \). Then, a \( k \times k \)-permutation clique is a clique of size \( k \) in \( G \) that contains exactly one vertex from each row and exactly one vertex from each column. In other words, there exists a permutation \( \pi \) on \([k] \) such that the vertices of the clique are \((1, \pi(1)), \ldots, (k, \pi(k))\). The task is to decide whether there exists a \( k \times k \)-permutation clique in \( G \). Lokshtanov et al. [29] proved that there is no \( \mathcal{O}^*(2^{o(k \log k)}) \)-time algorithm for \( k \times k \) Permutation Clique, unless \( \text{ETH} \) fails.

Let \((G, k)\) be an instance of \( k \times k \) Permutation Clique. We create an agent \( a_i \) for each row \( i \in [k] \), and an item \( b_j \) for each column \( j \in [k] \). We construct an instance of \( \alpha \)-GLOBALLY OPTIMAL ASSIGNMENT consisting of \( k^2 \) layers, each corresponds to a row-column pair \((i, j)\), containing the preference profile \( P_{(i,j)} \) defined as follows.

- \( a_i : b_j \)
- \( a_r : \{b_q \mid \{(i,j),(r,q)\} \in E(G), q \neq j\} \) (sorted in ascending order by \( q \))

\( \forall r \in [k] \setminus \{i\} \).

We finally set \( \alpha = k \). We now prove that there exists a \( k \times k \)-permutation clique in \( G \) if and only if there exists a \( k \)-globally optimal assignment for the constructed instance.

(\( \Rightarrow \)) Suppose there exists a permutation \( \pi \) for \([k] \) such that \((1, \pi(1)), \ldots, (k, \pi(k))\) form a clique in \( G \). We define an assignment \( p \) by
p(a_i) = b_{\pi(i)} for each i \in [k], and we claim that p is pareto optimal in each $P_{(i,\pi(i))}$ (see Figure 3). Observe that for each i \in [k], $b_{\pi(i)}$ is acceptable by $a_i$ in $P_{(i,\pi(i))}$ and in all profiles $P_{(j,\pi(j))}$ such that $j \in [k] \setminus \{i\}$ since there is an edge between (i, \pi(i)) and each (j, \pi(j)). Moreover, each $P_{(j,\pi(j))}$ contains no self loops due to the fact that all the items are allocated. Since we sorted each preference list in an ascending order by the item indices, all the agents respect each other in each preference profile and by Lemma 3, p does not admit trading cycles in any layer.

(\Leftarrow) Suppose there exists a k-globally optimal assignment p for the constructed instance. Note that if p is pareto optimal in some profile $P_{(i,j)}$, it must satisfy $p(a_i) = b_j$, as otherwise, $a_i$ would admit a self loop ($b_j$ is not acceptable by any agent but $a_i$ in $P_{(i,j)}$). Hence, we have that for each i \in [k], p is pareto optimal in at most one profile among $P_{(i,1)}, \ldots, P_{(i,k)}$ and in at most one profile among $P_{(1,i)}, \ldots, P_{(k,i)}$. Since we set $\alpha = k$, we have that for each i \in [k], p is pareto optimal in exactly one profile among $P_{(1,\pi(1))}, \ldots, P_{(i,\pi(i))}$ and in exactly one profile among $P_{(1,i)}, \ldots, P_{(k,i)}$. This implies that there exists a permutation $\pi$ on [k] such that p is pareto optimal in $P_{(i,\pi(i))}$ for each i \in [k].

We claim that \{(i, \pi(i)) | i \in [k]\} is the vertex set of a clique in G. Towards a contradiction, suppose that there exist two different rows $i_1$ and $i_2$ such that \{(i_1, \pi(i_1)), (i_2, \pi(i_2))\} \notin E(G). By the construction of the preference lists, observe that $b_{\pi(i_2)}$ is not acceptable by $a_{i_2}$ in $P_{(i_1,\pi(i_1))}$. Therefore, p is not a legal assignment for $P_{(i_1,\pi(i_1))}$, a contradiction to its optimality.

It holds that $n+m+\alpha = O(k)$. Thus, by Proposition $5$, we conclude that there is no $O^*(2^{\alpha(\log k)})$-time algorithm for $\alpha$-GLOBALLY OPTIMAL ASSIGNMENT, unless ETH fails.

We now prove a that the same result holds also for the parameter $n + m + (\ell - \alpha)$. 

\[
\text{Fig. 3.} \text{ For the presented graph, the constructed instance of } \alpha\text{-GLOBALLY OPTIMAL ASSIGNMENT consists of 9 layers, one for each row-column pair; } \{(1, 1), (2, 3), (3, 2)\} \text{ is a } 3 \times 3\text{-permutation clique. Thus, the assignment } p \text{ which satisfies } p(a_1) = b_1, p(a_2) = b_2, \text{ and } p(a_3) = b_2 \text{ is } 3\text{-globally optimal for the constructed instance, and is pareto optimal in the profiles } P_{(1,1)}, P_{(2,3)}, \text{ and } P_{(3,2)}.
\]
Theorem 6. Unless ETH fails, there does not exists an algorithm for α-
Globally Optimal Assignment with running time $O^*(2^\alpha \log k)$, where
$k = n + m + (\ell - \alpha)$.

Proof. We provide a different linear parameter reduction from $k \times k$ Perm-
utation Clique for the parameter $n + m + (\ell - \alpha)$. Let $(G = (V, E), k)$ be an
instance of $k \times k$ Permutation Clique (recall that $V = [k] \times [k]$). We cre-
tate $2k$ agents $a_1, \ldots, a_k, c_1, \ldots, c_k$, and $2k$ items $b_1, \ldots, b_k, d_1, \ldots, d_k$. Then,
we construct an instance of α-Globally Optimal Assignment consisting of
$\ell = 1 + k + \binom{k^2}{2} - |E|$ layers as follows. Intuitively, the first layer requires the
agents $a_1, \ldots, a_k$ to accept only the items $b_1, \ldots, b_k$, and the agents $c_1, \ldots, c_k$ to
accept only the items $d_1, \ldots, d_k$. Formally, it is defined as follows:

- $a_i : b_1 > \ldots > b_k \, \forall i \in [k]$
- $c_i : d_1 > \ldots > d_k \, \forall i \in [k]$

The next $k$ layers require $a_i$ and $c_i$ to “agree” with each other. Namely,$p(a_i) = b_j$ if and only if $p(c_i) = d_j$. To this end, for each $j \in [k]$, we construct the following preference profile:

- $a_i : \{b_q \mid q \in [k]\} \setminus \{b_j\}$ (sorted in ascending order by $q$) $\{d_q \mid q \in [k]\} \setminus b_j \, \forall i \in [k]$
- $c_i : d_j > b_j > \{d_q \mid q \in [k]\} \setminus \{d_j\}$ (sorted in ascending order by $q$) $\forall i \in [k]$

We construct additional $\binom{k^2}{2} - |E|$ layers (notice that there are exactly
$\binom{k^2}{2} - |E|$ pairs of vertices $\{(i_1, j_1), (i_2, j_2)\}$ that are not adjacent in $G$). Each
layer is defined with respect to a vertex pair $(i_1, j_1) \neq (i_2, j_2)$ such that
$\{(i_1, j_1), (i_2, j_2)\} \notin E$ as follows:

- $a_{i_1} : \{b_q \mid q \in [k]\} \setminus \{b_{j_1}\}$ (sorted in ascending order by $q$) $d_{j_2} > b_{j_1}$
- $c_{i_2} : \{d_{i_2} \mid q \in [k]\} \setminus \{d_{j_2}\}$ (sorted in ascending order by $q$) $b_{j_1} > d_{j_2}$
- $a_i : \{b_q \mid q \in [k]\} \setminus \{b_{j_1}\}$ (sorted in ascending order by $q$) $b_{j_1} \forall i \in [k] \setminus \{i_1\}$
- $c_i : \{d_q \mid q \in [k]\} \setminus \{d_{j_2}\}$ (sorted in ascending order by $q$) $d_{j_2} \forall i \in [k] \setminus \{i_2\}$
Finally, we set \( \alpha = \ell \). The reduction can be done in a polynomial time using simple polynomial-time operations on graphs. Before we prove the correctness of the construction, let us prove the following:

**Claim 2.** Let \( p \) be an \( \ell \)-globally optimal assignment for the constructed instance. Then, for each \( i \in [k] \), \( p(a_i) \in \{ b_1 \ldots b_k \} \), and \( p(c_i) \in \{ d_1, \ldots, d_k \} \).

**Proof.** Observe that in the first layer, \( b_1, \ldots, b_k \) are all only acceptable by \( a_1, \ldots, a_k \) and \( d_1, \ldots, d_k \) are all only acceptable by \( c_1, \ldots, c_k \). Hence, \( p \) must allocate all the items. Towards a contradiction, suppose that there exists \( b_j \) that is not assigned to any agent (the case when there exists \( d_j \) that is not assigned to any agent is symmetric). This implies that there exists an agent \( a_i \) that satisfies \( p(a_i) = b_0 \) and therefore admits a self loop with \( b_j \), a contradiction to the optimality of \( p \) in the first layer. \( \square \)

**Claim 3.** Let \( p \) be an \( \ell \)-globally optimal assignment for the constructed instance. Then, for every \( i, j \in [k] \), \( p(a_i) = b_j \) if and only if \( p(c_i) = d_j \).

**Proof.** Towards a contradiction, suppose there exist \( i, j \in [k] \) such that \( p(a_i) = b_j \) and \( p(c_i) \neq d_j \) (the second case where \( p(c_i) = d_j \) and \( p(a_i) \neq b_j \) is symmetric). By Claim 2, \( p(c_i) = d_q \) such that \( q \neq j \). Note that \( p \) admits the trading cycle \((a_i, b_j, c_i, d_q)\) in layer \( j \), a contradiction to the \( \ell \)-global optimality of \( p \). \( \square \)

**Claim 4.** Let \( p \) be an \( \ell \)-globally optimal assignment for the constructed instance. Let \( \pi : [k] \to [k] \) be defined as follows: \( \pi(i) = j \) such that \( p(a_i) = b_j \) for each \( i \in [k] \). Let \( V' = \{(i, \pi(i)) \mid i \in [k]\} \). Then \( V' \) is the vertex set of a \( k \times k \)-permutation clique in \( G \).

**Proof.** Observe that \( \pi \) is a well-defined permutation on \( [k] \) since \( p \) allocates all the items in \( \{b_1, \ldots, b_k\} \). Towards a contradiction, suppose there exist different \( i_1, i_2 \in [k] \) such that \( \{(i_1, \pi(i_1)), (i_2, \pi(i_2))\} \notin E \). First, by Claims 2 and 3, \( p(c_{i_1}) = d_{\pi(i_1)} \) and \( p(c_{i_2}) = d_{\pi(i_2)} \). Second, by the construction of the instance, there exists a layer among the \( \binom{k}{2} - |E| \) last layers which corresponds to the non-adjacent pair \( \{(i_1, \pi(i_1)), (i_2, \pi(i_2))\} \). Note that \( p \) admits the trading cycle \((a_{i_1}, b_{\pi(i_1)}, c_{i_2}, d_{\pi(i_2)})\) in this layer, a contradiction. \( \square \)

We now prove that there exists a \( k \times k \)-permutation clique in \( G \) if and only if there exists an \( \ell \)-globally optimal assignment for the constructed instance.

(\( \Rightarrow \)): Let \( \pi \) be a permutation on \( [k] \) such that \( V' = \{(1, \pi(1)), \ldots, (k, \pi(k))\} \) is the vertex set of a \( k \times k \)-permutation clique in \( G \). Let us define an assignment \( p \) by \( p(a_i) = b_{\pi(i)} \) and \( p(c_i) = d_{\pi(i)} \) for each \( i \in [k] \). Observe that both \( \{a_1, \ldots, a_k\} \) and \( \{c_1, \ldots, c_k\} \) contain agents that pairwise respect each other. Since each \( d_j \) is not acceptable by any \( a_i \), and each \( b_j \) is not acceptable by any \( c_i \), we have by Lemma 3 that \( p \) does not admit trading cycles in the first layer. Moreover, since all the items are allocated, it does not admit self loops in the first layer, implying that \( p \) is pareto optimal in the this layer. Furthermore, \( p \) is also pareto optimal in the next \( k \) layers since \( a_i \) and \( c_i \) “agree” with each other for each
i \in [k]. Formally, for every j \in [k], both \{a_1, \ldots, a_k\} and \{c_1, \ldots, c_k\} contain agents that pairwise respect each other in layer 1 + j. By the construction of layer 1 + j, and by Lemma 3, the only possible trading cycle in this layer occurs when p(a_i) = b_j, and p(c_i) = d_q such that q \neq j. We claim that p is also pareto optimal in the last \(k^2 - |E|\) layers. Suppose, for the sake of contradiction, that there exists a pair \((q_1, r_1), (q_2, r_2)\) \notin E, such that p is not pareto optimal in its corresponding layer. By Lemma 3, the only possible trading cycle in this layer is \((a_{q_1}, b_{r_1}, c_{q_2}, d_{r_2})\). Thus, we have that p(a_{q_1}) = b_{r_1} and p(c_{q_2}) = d_{r_2}. By the construction of p, both \((q_1, r_1)\) and \((q_2, r_2)\) are vertices in \(V'\), a contradiction.

(⇐): Let p be an \(\ell\)-globally optimal assignment for the constructed instance. By Claim \[4\] there exists a \(k \times k\)-permutation clique in \(G\). \QEDB

Theorem 7. \(\alpha\)-Globally Optimal Assignment is \(W[1]\)-hard for the parameter \(k = m + \alpha\).

Proof. We provide a parameterized (and also polynomial time) reduction from the \(W[1]\)-hard problem Multicolored Independent Set to \(\alpha\)-Globally Optimal Assignment. The input of Multicolored Independent Set consists of an undirected graph \(G = (V, E)\), and a coloring \(c : V \rightarrow [k]\) that colors the vertices in \(V\) with \(k\) colors. The task is to decide whether \(G\) admits a multicolored independent set of size \(k\), which is an independent set (i.e. a vertex subset with pair-wise non-adjacent vertices) \(V' \subseteq V\) that satisfies \(\{c(v') \mid v' \in V'\} = [k]\) and \(|V'| = k\).

Given an instance \((G = (V, E), c)\) of Multicolored Independent Set, denote \(V = \{v_1, \ldots, v_n\}\). We construct an instance of \(\alpha\)-Globally Optimal Assignment consisting of \(n\) layers, with agent set \(A = \{a_1, \ldots, a_n\}\), and item set \(I = \{b_1, \ldots, b_k\}\). Intuitively, each agent \(a_i\) corresponds to the vertex \(v_i\), and each item \(b_i\) corresponds to the color \(i \in [k]\). We construct the instance such that the items are allocated to agents whose corresponding vertices form a multicolored independent set. For each \(i \in [n]\), we construct the preference profile in layer \(i\), denoted by \(P_i\), as follows:

- \(a_i : b_{c(v_i)}\)
- \(a_j : \emptyset \forall j \in [n] \setminus \{i\}\) and \(c(v_j) = c(v_i)\)
- \(a_j : \emptyset \forall j \in [n] \setminus \{i\}\) such that \(\{v_i, v_j\} \in E\) and \(c(v_j) \neq c(v_i)\)
- \(a_j : b_{c(v_j)} \forall j \in [n] \setminus \{i\}\) such that \(\{v_i, v_j\} \notin E\) and \(c(v_j) \neq c(v_i)\)

Finally, we set \(\alpha = k\). The construction clearly can be done in a polynomial time. Before we prove the correctness of the reduction, let us prove the following claim.

Claim 5. Let \(p : A \rightarrow I \cup \{b_\emptyset\}\) be an assignment. Then for any \(i \in [n]\), if \(p\) is pareto optimal in layer \(i\), then it satisfies that \(p(a_i) = b_{c(v_i)}\), and for each \(j \in [n] \setminus \{i\}\) such that \(\{v_i, v_j\} \in E\), \(p(a_j) = b_\emptyset\).
Parameterized Analysis of Assignment Under Multiple Preferences 23

Fig. 4. In the presented graph colored using 5 colors, \{v_1, v_3, v_4, v_5, v_6\} is a multicolored independent set of size 5. The constructed instance contains 8 layers, one for each vertex; and the presented assignment is 5-globally optimal since it is pareto optimal in layers 1, 3, 4, 5, and 6.

Proof. Let \(i \in [n]\). Assume that \(p\) is pareto optimal in layer \(i\). Note that \(a_i\) is the only agent that accepts \(b_{c(v_i)}\) in \(P_i\) and this is the only item it accepts. Hence, \(p\) must allocate \(b_i\) to \(a_i\), as otherwise, \(a_i\) would be a part of a self loop. Moreover, for each \(j \in [n]\) such that \(\{v_i, v_j\} \in E\), the preference list of \(a_j\) in \(P_i\) is empty, implying that \(p(a_j) = b_\emptyset\). ⊓⊔

We now prove that \(G\) admits a multicolored independent set of size \(k\) if and only if the constructed instance admits a \(k\)-globally optimal assignment.

(⇒) Assume that \(G\) admits a multicolored independent set \(V' = \{v_{i_1}, \ldots, v_{i_k}\}\). We define an assignment \(p\) by \(p(a_{i_j}) = b_{c(v_{i_j})}\) for each \(j \in [k]\) (see Figure 4), and \(p(a_q) = b_\emptyset\) for each \(q \notin \{i_1, \ldots, i_k\}\). Note that the assignment does not assign the same item to two or more agents since the colors of the vertices in \(V'\) are distinct. We claim that \(p\) is \(k\)-globally optimal for the constructed instance. First, the agents \(a_{i_1}, \ldots, a_{i_k}\) contain non-empty preference lists in the preference profiles \(P_{i_1}, \ldots, P_{i_k}\) since their corresponding vertices are pair-wise non-adjacent and colored with distinct colors. Hence, each item is acceptable by its owner in each \(P_{i_j}\). Second, \(P_{i_1}, \ldots, P_{i_k}\) cannot include trading cycles since each preference list is of length at most one.

(⇐) Suppose there exists a \(k\)-globally optimal assignment \(p\) for the constructed instance. Then, there exist \(k\) layers \(i_1, \ldots, i_k\) such that \(p\) is pareto optimal in each \(P_{i_j}\). Proposition 5 implies that for each \(j \in [k]\), \(p(a_{i_j}) = b_{c(v_{i_j})}\), and for each \(q \in [n]\) such that \(\{v_q, v_j\} \in E\), \(p(a_q) = b_\emptyset\). We claim that \(V' = \{v_{i_1}, \ldots, v_{i_k}\}\) is a multicolored independent set. First, we show that it forms
such that \( \{v_{i_1}, v_{i_2}\} \in E \). Then, the preference list of \( a_{i_1} \) in \( P_{v_{i_2}} \) is empty, and therefore \( p \) is not legal in \( P_{v_{i_2}} \), a contradiction to its pareto optimality in \( P_{v_{i_2}} \).

Second, we show that it is multicolored. Towards a contradiction, suppose there exist \( j_1, j_2 \in [k] \) such that \( c(v_{i_{j_1}}) = c(v_{i_{j_2}}) \). By Claim [5], since \( p \) is pareto optimal in layers \( i_{j_1} \) and \( i_{j_2} \), it satisfies \( p(a_{i_{j_1}}) = b_{c(v_{i_{j_1}})} \) and \( p(a_{i_{j_2}}) = b_{c(v_{i_{j_2}})} \). We have that \( p \) assigns \( b_{c(v_{i_{j_1}})} \) to two agents, a contradiction. 

We now provide a similar hardness result for the parameter \( m + (\ell - \alpha) \).

**Theorem 8.** \( \alpha \)-Globally Optimal Assignment is \( \text{W}[1] \)-hard for the parameter \( k = m + (\ell - \alpha) \).

**Proof.** We provide a different parameterized (and polynomial time) reduction from Multicolored Independent Set to \( \alpha \)-Globally Optimal Assignment.

Given an instance \((G = (V, E), c)\) of Multicolored Independent Set, assume that \( V = \{v_1, \ldots, v_n\} \). We construct an instance of \( \alpha \)-Globally Optimal Assignment with agent set \( A = \{a_1, \ldots, a_n\} \) and item set \( I = \{b_1, \ldots, b_\ell\} \), consisting of \( \ell = n + 1 \) layers. Informally speaking, each agent \( a_i \) corresponds to a vertex \( v_i \), and each item \( b_c \) corresponds to a color \( c \). We construct an instance such that the agents that allocate items from \( I \) in an \( \ell \)-globally optimal assignment correspond to vertices which form a multicolored independent set in \( G \). Moreover, we require each agent to allocate either the item that corresponds to its color with respect to \( c \), or \( b_\ell \). The first layer is defined as follows:

\[
\bullet \quad a_i : b_{c(i)} \quad \forall i \in [k]
\]

We construct \( n \) additional layers. For each \( i \in [n] \), layer \( 1 + i \) is defined as follows:

\[
\bullet \quad a_i : \{b_j \mid j \in [k] \setminus \{b_{c(v_j)}\} \} \text{ (ordered arbitrarily)} > b_{c(v_i)} \\
\bullet \quad a_j : b_{c(v_j)} \quad \forall j \in [n] \setminus \{i\} \text{ such that (1) } \{v_i, v_j\} \in E \text{ and } c(v_j) = c(v_i) \text{ or (2) } \{v_i, v_j\} \notin E \\
\bullet \quad a_j : b_{c(v_j)} > b_{c(v_j)} \quad \forall j \in [n] \setminus \{i\} \text{ such that } \{v_i, v_j\} \in E \text{ and } c(v_j) \neq c(v_i)
\]

We finally set \( \alpha = \ell \). We claim that \( G \) admits a multicolored independent set of size \( k \) if and only if there exists an \( \ell \)-globally optimal assignment for the constructed instance.

\((\Rightarrow)\): Suppose \( G \) admits a multicolored independent set of size \( k \), \( V' = \{v_{i_1}, \ldots, v_{i_k}\} \). We define an assignment \( p \) by \( p(a_{i_j}) = b_{c(v_{i_j})} \) for each \( j \in [k] \),
Parameterized Analysis of Assignment Under Multiple Preferences

and \( p(a_i) = b_9 \) for each \( i \notin \{i_1, \ldots, i_k\} \). We claim that \( p \) is pareto optimal in all the layers. First, observe that for each \( i \in [k] \), \( p(a_i) \) is acceptable by \( a_i \) in each layer, and each layer cannot admit self loops since all the items are allocated. Second, \( p \) is pareto optimal in the first layer: since each agent accepts only a single item, no subset of agents can admit a trading cycle. Third, we claim that \( p \) is pareto-optimal in layer \( 1 + i \) for each \( i \in [n] \). Towards a contradiction, suppose that there exists \( i \in [n] \) such that \( p \) is not pareto optimal in layer \( 1 + i \). By Proposition \( \Box \), there exists a trading cycle in layer \( 1 + i \). Observe that a trading cycle in layer \( 1 + i \) cannot contain agents with a single item on their preference list. Moreover, it cannot contain only agents from \( \{a_j \mid \{v_j, v_r\} \in E \text{ and } c(v_j) \neq c(v_r)\} \) since each such agent ranks \( b_{c(v_r)} \) first in its preference list. Therefore, we have that the only possible trading cycle in layer \( 1 + i \) consists of the agent \( a_i \) and an agent \( a_r \) such that \( p(a_i) = b_{c(v_r)}, c(v_i) \neq c(v_r), \{v_i, v_r\} \in E \), and \( p(a_r) = b_{c(v_i)} \). Then, by the construction of \( p \), we have that \( v_i, v_r \in V' \), a contradiction.

\((\Leftarrow)\): Suppose there exists an \( \alpha \)-globally optimal assignment \( p \) for the constructed instance. We first show that \( p \) must allocate all the items. Towards a contradiction, assume that there exists \( j \in [k] \) such that \( b_j \) is not assigned to any agent. Since \( n \geq k \), there exists an agent \( a_t \) that is not assigned to an item. Notice that \( a_t \) accepts all the items in layer \( 1 + t \), thus, \( a_t \) and \( b_j \) admit a self loop, a contradiction. Let \( a_{i_1}, \ldots, a_{i_k} \) denote the agents that are allocated an item from \( I \). We claim that \( p(a_{i_j}) = b_{c(v_{i_j})} \) for each \( j \in [k] \). For the sake of contradiction, assume that there exists \( j \in [k] \) such that \( p(a_{i_j}) \neq b_{c(v_{i_j})} \). We consider two cases: (1) \( p(a_{i_j}) = b_9 \). In this case, \( p \) admits a self loop among \( a_{i_j} \) and \( b_{c(v_{i_j})} \) in the first layer, a contradiction. (2) \( p(a_{i_j}) = b_r \neq b_{c(v_{i_j})} \). In this case, \( p \) is not legal in the first layer, a contradiction. This implies that the vertices in the set \( V' = \{v_{i_1}, \ldots, v_{i_k}\} \) are colored with all the colors. We claim that \( V' \) is a also an independent set. Towards a contradiction, assume there exist \( j, r \in [k] \) such that \( \{v_{i_j}, v_{i_r}\} \in E \). Then we have that \( a_{i_j} \) and \( a_{i_r} \) form the trading cycle \((a_{i_j}, b_{c(v_{i_j})}, a_{i_r}, b_{c(v_{i_r})}) \) in layer \( 1 + i_r \), a contradiction to Proposition \( \Box \).

\( \square \)

4 NP-Hardness

In this section, we prove that for any \( 2 \leq \alpha \leq \ell \), \( \alpha \)-Globally Optimal Assignment is \( \text{NP} \)-hard. After that, we provide three constructions of preference profiles given an instance of \( 3 \)-SAT. Then, we will rely on these constructions to prove that \( \alpha \)-Globally Optimal Assignment is \( \text{para-NP} \)-hard with respect to the parameter \( \ell + d \).

Theorem 9. For any \( 2 \leq \alpha \leq \ell \), \( \alpha \)-Globally Optimal Assignment is \( \text{NP} \)-hard.

Proof. We rely on a reduction made by Aziz et al. [6] from the Serial Dictatorship Feasibility problem, which was proved to be \( \text{NP} \)-hard by Saban and Sethuraman [31]. In the Serial Dictatorship Feasibility problem, the input is a tuple \((A, I, P, a, b)\) where \( A \) is a set of \( n \) agents, \( I \) is a set of \( n \) items, \( P \) is the
preferences profile in which each agent has a complete linear order on the items, \(a \in A\), and \(b \in I\). The task is to decide whether there exists a permutation for which serial dictatorship (defined in Section 2) allocates item \(b\) to agent \(a\). Given such \((A,I,P,a,b)\), Aziz et al. [6] constructed two preference profiles, \(P_1\) and \(P_2\), such that \((A,I,P,a,b)\) is a \textit{Yes}-instance if and only if there exists an assignment that is pareto optimal in both \(P_1\) and \(P_2\).

Let \((A,I,P,a,b)\) be an instance of \textbf{Serial Dictatorship Feasibility}. We construct the aforementioned preference profiles \(P_1\) and \(P_2\). We also add \(\ell - \alpha\) additional new items \(c_1,\ldots,c_{\ell - \alpha}\), and define \(I' = I \cup \{c_1,\ldots,c_{\ell - \alpha}\}\). We construct \(\ell - \alpha\) preference profiles \(P'_1,\ldots,P'_{\ell - \alpha}\), where for each \(i \in [\ell - \alpha]\), \(P'_i\) is defined as follows:

- \(a : c_i\)
- \(a' : \emptyset \forall a' \in A \setminus \{a\}\)

In other words, the only item that agent \(a\) accepts in \(P'_i\) is \(c_i\), and for each \(a' \in A \setminus \{a\}\), \(a'\) accepts no items. We construct an instance of \textbf{\(\alpha\)-Globally Optimal Assignment} with the agent set \(A\), and the item set \(I'\), consisting of \(\ell\) layers. The first two layers contain the preference profiles \(P_1\) and \(P_2\), the next \(\alpha - 2\) layers contain copies of \(P_1\), and the next \(\ell - \alpha\) layers contain \(P'_1,\ldots,P'_{\ell - \alpha}\).

Let us first prove the following.

\textbf{Claim 6.} Let \(p\) be an \(\alpha\)-globally optimal assignment for the constructed instance. Then \(p\) is pareto optimal in \(P_1\) and \(P_2\).

\textbf{Proof.} Note that the only pareto optimal assignment for \(P'_i\) is the assignment that allocates \(c_i\) to \(a\), and \(b_0\) to each \(a' \in A \setminus \{a\}\). Hence, there does not exist an assignment that is pareto optimal in both \(P'_i\) and \(P'_j\) when \(i \neq j\). Moreover, there does not exist an assignment that is pareto optimal in both \(P'_i\) and \(P_1\) or in both \(P'_i\) and \(P_2\) since \(c_i\) is not acceptable by \(a\) in \(P_1\) as well as \(P_2\). The only possible option is that \(p\) is pareto optimal in the first \(\alpha\) layers, then we have that \(p\) is pareto optimal in both \(P_1\) and \(P_2\).

We claim that there exists an \(\alpha\)-globally optimal assignment for the constructed instance if and only if \((A,I,P,a,b)\) is a \textit{Yes}-instance.

\((\Rightarrow)\): Suppose there exists an \(\alpha\)-globally optimal assignment \(p\) for the constructed instance. By Claim 6, \(p\) is pareto optimal in both \(P_1\) and \(P_2\). By Aziz et al. [6], this implies that \((A,I,P,a,b)\) is a \textit{Yes}-instance.

\((\Leftarrow)\): Assume that \((A,I,P,a,b)\) is a \textit{Yes}-instance. By Aziz et al. [6], there exists an assignment \(p\) that is pareto optimal in both \(P_1\) and \(P_2\). By the construction of the instance, \(p\) is pareto optimal in the first \(\alpha\) layers, hence \(p\) is \(\alpha\)-globally optimal for the constructed instance. \(\square\)

\textbf{Corollary 5.} \(\alpha\)-\textbf{Globally Optimal Assignment} is \textit{para-NP-hard} for the parameter \(\ell + \alpha\).
In contrast to Theorem 9, we mention that it is impossible to obtain the same hardness result when $\alpha = 1$.

**Observation 6.** $\alpha$-Globally Optimal Assignment is solvable in a polynomial time when $\alpha = 1$.

**Proof.** We can simply find a pareto optimal assignment for the first layer using Corollary 1. $\square$

We define three constructions of preference profiles given an instance of 3-SAT, and we consider their connections to the satisfiability of the formula. We will rely on these connections to design a polynomial reduction from 3-SAT to $\alpha$-Globally Optimal Assignment that shows that $\alpha$-Globally Optimal Assignment is para-NP-hard with respect to the parameter $\ell + \alpha + d$ (where $d$ is the maximal length of a preference list). We will also rely on these results in the design of our cross-compositions, which prove that $\alpha$-Globally Optimal Assignment does not admit a polynomial kernel with respect to the parameters $n + m + \alpha$, $n + m + (\ell - \alpha)$, and $m + \ell$ unless $\text{NP} \subseteq \text{coNP/poly}$.

Let $n, m \in \mathbb{N}$ be positive integers. We use the notation $A(m, n)$ to refer to the agent set $A(m, n) = \{a_{i,j} \mid i \in [m], j \in [n]\} \cup \{\overline{a}_{i,j} \mid i \in [m], j \in [n]\}$, and we use the notation $I(m, n)$ to refer to the item set $I(m, n) = \{b_{i,j} \mid i \in [m], j \in [n]\} \cup \{\overline{b}_{i,j} \mid i \in [m], j \in [n]\}$. We will define two preference profiles over $A(m, n)$ and $I(m, n)$: $P_1(m, n)$ and $P_2(m, n)$. Intuitively, given a 3-SAT instance with $n$ variables and $m$ clauses, we will construct the sets $A(m, n)$ and $I(m, n)$, which contain two agents and two items for each clause-variable pair. The way that these agents and items are assigned to each other in an assignment that is pareto optimal in both $P_1(m, n)$ and $P_2(m, n)$ will encode a boolean assignment for the variable set of the instance. We define the preference profile $P_1(m, n)$ as follows.

- $a_{i,j} : b_{i,j} > \overline{b}_{i,j}$ $\forall i \in [m], j \in [n]$
- $\overline{a}_{i,j} : b_{i,j} > \overline{b}_{i,j}$ $\forall i \in [m], j \in [n]$
Claim 7. An assignment $p : A(m, n) \rightarrow I(m, n) \cup \{b_0\}$ is pareto optimal in $P_1(m, n)$ if and only if $\{p(a_{i,j}), p(\overline{a_{i,j}})\} = \{b_i, \overline{b_i}\}$ for each $i \in [m]$ and $j \in [n]$.

Proof. $(\Rightarrow)$: Assume that $p$ is pareto optimal in $P_1(m, n)$. Observe that $b_i, \overline{b_i}$ are only acceptable by $a_{i,j}$ and $\overline{a_{i,j}}$. If $\left|\{p(a_{i,j}), p(\overline{a_{i,j}})\} \cap \{b_i, \overline{b_i}\}\right| < 2$, then at least one of the items $b_i, \overline{b_i}$ does not have an owner; this implies that at least one of the agents $a_{i,j}, \overline{a_{i,j}}$ admits a self loop, a contradiction to Proposition 1 (see Figure 5).

$(\Leftarrow)$: Assume that $\{p(a_{i,j}), p(\overline{a_{i,j}})\} = \{b_i, \overline{b_i}\}$ for each $i \in [m]$ and $j \in [n]$. Observe that the agents in $A(m, n)$ pairwise respect each other in $P_1(m, n)$; then, by Lemma 3 $p$ does not admit trading cycles in $P_1(m, n)$. Since $p$ allocates all the items, it also does not admit self loops. Thus, by Proposition 1 $p$ is pareto optimal in $P_1(m, n)$.

We now define the second preference profile $P_2(m, n)$ over $A(m, n)$ and $I(m, n)$. Intuitively, being pareto optimal in both $P_1(m, n)$ and $P_2(m, n)$ requires the agents to “agree” with each other. That is, $a_{i,j}$ gets $b_{i,j}$ if and only if $a_{i,\overline{j}}$ gets $b_{i,\overline{j}}$ for each $i \neq i_2$. For $j \in [n]$, we denote the sets $P_j^{\text{true}}$, and $P_j^{\text{false}}$ by $P_j^{\text{true}} = \{(a_{i,j}, b_{i,j}), (\overline{a_{i,j}}, \overline{b_{i,j}}) \mid i \in [m]\}$, and $P_j^{\text{false}} = \{(a_{i,j}, b_{i,j}), (\overline{a_{i,j}}, b_{i,j}) \mid i \in [m]\}$. Informally speaking, $P_j^{\text{true}}$ and $P_j^{\text{false}}$ will correspond to setting the variable $x_j$ to true or false, respectively.

- $a_{m,j} : b_{m,j} > b_{m-1,j} > \overline{b_{m,j}} \forall j \in [n]$
- $a_{m,j} : b_{m,j} > b_{m-1,j} > \overline{b_{m,j}} \forall j \in [n]$
- $a_{i,j} : b_{i-1,j} > b_{i,j} > b_{i+1,j} > b_{i,j} \forall i \in \{2, \ldots, m-1\}, j \in [n]$
- $a_{i,j} : b_{i-1,j} > b_{i,j} > b_{i+1,j} > b_{i,j} \forall i \in \{2, \ldots, m-1\}, j \in [n]$
- $a_{i,j} : b_{i,j} > b_{i,j} > b_{i,j} \forall j \in [n]$
- $a_{i,j} : b_{i,j} > b_{i,j} > b_{i,j} \forall j \in [n]$

Claim 8. An assignment $p : A(m, n) \rightarrow I(m, n) \cup \{b_0\}$ is pareto optimal in both $P_1(m, n)$ and $P_2(m, n)$ if and only if for each $j \in [n]$, either $P_j^{\text{true}} \subseteq p$ or $P_j^{\text{false}} \subseteq p$.

Proof. $(\Rightarrow)$: Assume that $p$ is pareto optimal in both $P_1(m, n)$ and $P_2(m, n)$. Towards a contradiction, suppose that there exists $j \in [n]$ satisfying that both $P_j^{\text{true}} \not\subseteq p$ and $P_j^{\text{false}} \not\subseteq p$. By Claim 7, there exist $i_1, i_2 \in [m]$ such that $i_1 < i_2$, satisfying that either (1) $p(a_{i_1,j}) = b_{i_1,j}$ and $p(a_{i_2,j}) = \overline{b_{i_2,j}}$, or (2) $p(a_{i_1,j}) = \overline{b_{i_1,j}}$ and $p(a_{i_2,j}) = b_{i_2,j}$. For the first case, note that there must exist $i \in [m]$ such that $i_1 \leq i < i_2$, $p(a_{i,j}) = b_{i,j}$, and $p(a_{i+1,j}) = b_{i+1,j}$. Then we have that $p$ admits the trading cycle $(a_{i,j}, b_{i,j}, a_{i+1,j}, b_{i+1,j})$ in $P_2(m, n)$, a contradiction. For the second case, we have that there exists $i \in [m]$ such that $i_1 \leq i < i_2$, $p(a_{i,j}) = b_{i,j}$, and $p(a_{i+1,j}) = \overline{b_{i+1,j}}$. By Claim 7, $p(a_{i,j}) = b_{i,j}$ and $p(\overline{a_{i+1,j}}) = \overline{b_{i+1,j}}$, then $p$ admits the trading cycle $(\overline{a_{i+j}}, b_{i,j}, \overline{a_{i+1,j}}, \overline{b_{i+1,j}})$ in $P_2(m, n)$, a contradiction.
Let \( D = (X, C) \) be an instance of 3-SAT where \( X = \{x_1, \ldots, x_n\} \) is the set of variables, and \( C = \{C_1, \ldots, C_m\} \) is the set of clauses, each of size 3. In order to construct the third preference profile \( P_3(D) \), order the literals in each clause arbitrarily, such that \( C_i = \ell_1^i \lor \ell_2^i \lor \ell_3^i \) for each \( i \in [m] \). The third preference profile is responsible for the satisfiability of the formula. Let us define the following:

**Definition 10.** Let \( C_i = \ell_1^i \lor \ell_2^i \lor \ell_3^i \in C \). We define \( \text{ind}_D(i, j) \) as the index of the variable which appears in the \( j \)-th literal in \( C_i \) for each \( i \in [3] \).

For example, if \( C_1 = x_3 \lor \overline{x_2} \lor x_5 \), then \( \text{ind}_D(i, 1) = 3 \), \( \text{ind}_D(i, 2) = 1 \), and \( \text{ind}_D(i, 3) = 5 \). Briefly put, each assignment \( p \) that is pareto optimal in the profiles \( P_1(m, n), P_2(m, n) \), and \( P_3(D) \) contains either \( P_j^{\text{true}} = \{(a_{i,j}, b_{i,j}), (\overline{a_{i,j}}, b_{i,j})| i \in [m]\} \) or \( P_j^{\text{false}} = \{(a_{i,j}, b_{i,j}), (\overline{a_{i,j}}, b_{i,j})| i \in [m]\} \) for every \( j \in [n] \), and for each clause \( C_i = \ell_1^i \lor \ell_2^i \lor \ell_3^i \in C \), there exists at least one satisfied literal \( \ell_i^j \). Being pareto optimal in \( P_3(D) \) enforces \( a_{i,\text{ind}_D(i,j)} \) and \( a_{i,\text{ind}_D(i,j)} \) to admit a trading cycle if none of their corresponding literals is satisfied. We define the following:

**Definition 11.** For each \( i \in [m], \ j \in [3], \) we define \( b_D(i,j) = \begin{cases} b_{i,\text{ind}_D(i,j)} & \ell_i^j \text{ is negative} \\ b_{i,\text{ind}_D(i,j)} & \ell_i^j \text{ is positive} \end{cases} \).

Intuitively, when \( a_{i,\text{ind}_D(i,j)} \) gets \( b_D(i,j) \) and \( \overline{a_{i,\text{ind}_D(i,j)}} \) gets \( \overline{b_D(i,j)} \), it means that \( \ell_i \) is "satisfied". Preference profile \( P_3(D) \) is defined as follows:
Fig. 7. If \( p \) is pareto optimal in both \( P_1(m, n) \) and \( P_2(m, n) \), then every potential trading cycle in \( P_3(D) \) has this form; and it occurs when \( P_{\text{false}}^{i,1} \), \( P_{\text{false}}^{i,2} \), \( P_{\text{false}}^{i,3} \) \( \subseteq p \).

Claim 9. An assignment \( p : A(m, n) \rightarrow I(m, n) \cup \{b_0\} \) is pareto optimal in \( P_1(m, n) \), \( P_2(m, n) \), and \( P_3(D) \) if and only if:

1. For each \( j \in [n] \), either \( P_j^{\text{true}} \subseteq p \) or \( P_j^{\text{false}} \subseteq p \).
2. For each clause \( C_i = \ell_1^i \lor \ell_2^i \lor \ell_3^i \in \mathcal{C} \), there exists at least one \( j \in [3] \) such that \( p(a_{i,\text{ind}(i,j)}) = b_D(i, j) \).

Proof. (\( \Rightarrow \)): Assume that \( p \) is pareto optimal in \( P_1(m, n) \), \( P_2(m, n) \), and \( P_3(D) \). Claim 8 implies that \( p \) satisfies the first condition. We show that \( p \) also satisfies the second condition: Observe that every tuple of the form \((a_{i,\text{ind}(i,3)}, b_D(i,3), a_{i,\text{ind}(i,2)}, b_D(i,2), a_{i,\text{ind}(i,1)}, b_D(i,1))\) is a trading cycle in \( P_3(D) \) (see Figure 7). Then by Proposition 1 for each \( i \in [m] \), there must exist \( j \in [3] \) such that \( p(a_{i,\text{ind}(i,j)}) = b_D(i, j) \).

(\( \Leftarrow \)): Assume that \( p \) satisfies both conditions. By Claim 8, \( p \) is pareto optimal in both \( P_1(m, n) \) and \( P_2(m, n) \). We claim that \( p \) is also pareto optimal in \( P_3(D) \): First, \( p \) does not admit self loops in \( P_3(D) \) since all the items are allocated. Second, observe that every trading cycle in \( P_3(D) \) is of the form \((a_{i,\text{ind}(i,3)}, b_D(i,3), a_{i,\text{ind}(i,2)}, b_D(i,2), a_{i,\text{ind}(i,1)}, b_D(i,1))\). By the second condition (in each clause there exists at least one “satisfied” literal), \( p \) does not allow cycles of such form. Thus, we conclude that \( p \) pareto optimal in \( P_3(D) \).
Lemma 4. An instance $D = (X, C)$ of 3-SAT such that $|X| = n$ and $|C| = m$ is a Yes-instance if and only if there exists an assignment $p : A(m, n) \to I(m, n) \cup \{b\}$ that is pareto optimal in $P_3(m, n)$.

Proof. ($\Rightarrow$): Assume that $D$ is a Yes-instance. Then, there exists a boolean assignment $\varphi : X \to \{T, F\}$ that satisfies every clause in $C$. We construct an assignment $p : A(m, n) \to I(m, n) \cup \{b\}$ as follows. For each $x_j \in X$, if $\varphi(x_j) = T$, we add $P_j^{true}$ to $p$, and otherwise, we add $P_j^{false}$ to $p$. We claim that $p$ is pareto optimal in all three preference profiles. First, by Claim 9, $p$ is pareto optimal in $P_3(m, n)$ and in $P_2(m, n)$. Second, it is pareto optimal in the third layer: Since $\varphi$ satisfies every clause, we have that for each $C_i = \ell_i^1 \lor \ell_i^2 \lor \ell_i^3 \in C$, there exists $j \in [3]$ such that (1) if $\ell_i^j$ is positive, then $\varphi(i_{indj}(i, j)) = T$, and (2) if $\ell_i^j$ is negative, then $\varphi(i_{indj}(i, j)) = F$. By the definition of $b_D(i, j)$ and the construction of $p$, we have that $p(a_{i, j}) = b_D(i, j)$, and by Claim 9, $p$ is pareto optimal in $P_3(D)$.

($\Leftarrow$): Assume that there exists an assignment $p$ that is pareto optimal in $P_3(m, n)$, $P_2(m, n)$ and $P_3(D)$. By Claim 9, for each $j \in [n]$, either $P_j^{true} \subseteq p$ or $P_j^{false} \subseteq p$. We define a boolean assignment for $(X, C)$ by $\varphi(x_j) = T$ if $P_j^{true} \subseteq p$, and $\varphi(x_j) = F$ if $P_j^{false} \subseteq p$. By Claim 9, for each clause $C_i = \ell_i^1 \lor \ell_i^2 \lor \ell_i^3 \in C$, there exists $j \in [3]$ such that $p(a_{i, j}) = b_D(i, j)$. By the construction of $\varphi$, $\varphi(\ell_i^j) = T$. Hence, $\varphi$ satisfies each clause in $C$, and therefore $D$ is a Yes-instance.

We rely on these results to design a polynomial reduction from 3-SAT to $\alpha$-Globally Optimal Assignment.

Theorem 10. 3-SAT is polynomial-time reducible to $\alpha$-Globally Optimal Assignment where $\alpha = \ell = 3$ and $d = 3$ or where $\alpha = \ell = 4$ and $d = m$.

Proof. Given an instance $D = (X, C)$ of 3-SAT, such that $|X| = n$, and $|C| = m$, we construct an instance of $\alpha$-Globally Optimal Assignment with the agent set $A(m, n)$, and the item set $I(m, n)$, consisting of three layers. The first layer contains $P_1(m, n)$, the second contains $P_2(m, n)$, and the third contains $P_3(D)$, and we finally set $\alpha = \ell = 3$. The reduction clearly can be done in polynomial time. The correctness of the reduction is derived by Lemma 4 which implies that there exists a 3-globally optimal assignment for the constructed instance if and only if $D$ is a Yes-instance. Notice that we can add an additional layer to the constructed instance where the preference lists of all the agents are complete and equal to each other (in this case, $d = m$). In this case, all the agents respect each other and therefore each assignment that allocates all the items is pareto optimal. This implies that there exists a 4-globally optimal assignment for the constructed instance if and only if $D$ is a Yes-instance.

Since $\ell + d$ (or $\ell + (m - d)$) in the statement above is upper bounded by fixed constants, we conclude the following.

Corollary 6. $\alpha$-Globally Optimal Assignment is para-NP-hard with respect to the parameters $\ell + d$ and $\ell + (m - d)$. 
5 Non-Existence of Polynomial Kernels

Before we present our next results, let us provide the following observation on which we will rely in the next three cross-compositions.

Observation 7. Let $D = (X, C)$ be an instance of 3-SAT of size $n$. Then there exists an equivalent 3-SAT instance $U = (X_U, C_U)$ where $X_U = \{x_1, \ldots, x_n\}$ and $|C_U| = n$.

Proof. Since $D$ has size $n$, it has at most $n$ variables and at most $n$ clauses. Then there exists a one to one mapping from $X$ to $X_U$. We pick such arbitrary mapping and replace each variable in $D$ with its corresponding variable in $X_U$. We then append the clause set with copies of some existing clause until it has size $n$, and denote the resulting clause multi-set by $C_U$. The new instance $U = (X_U, C_U)$ is clearly equivalent to $D$. □

Theorem 11. There does not exist a polynomial kernel for $\alpha$-Globally Optimal Assignment with respect to the parameters $k_1 = n + m + \alpha$ and $k_2 = n + m + (\ell - \alpha)$, unless $\text{NP} \subseteq \text{coNP/poly}$.

Proof. We provide two cross-compositions from 3-SAT to $\alpha$-Globally Optimal Assignment. Given instances of 3-SAT $D_0 = (X_0, C_0), \ldots, D_{t-1} = (X_{t-1}, C_{t-1})$ of the same size $n \in \mathbb{N}$ for some $t \in \mathbb{N}$, we first modify each instance $D_i$ to have $X_i = \{x_1, \ldots, x_n\}$ and $|C_i| = n$ using Observation 7. The two algorithms construct instances of $\alpha$-Globally Optimal Assignment with $2n^2 + \lceil 2 \log t \rceil$ agents, $2n^2 + \lceil 2 \log t \rceil$ items, and $2 + t$ layers, that share the same agent set and item set (Shown in Figure 5). We first provide the first two layers, which are identical in both constructions, and then we provide the remaining $t$ layers for each construction separately. We create the agent sets $A(n, n) = \{a_{i,j} \mid i, j \in [n]\} \cup \{\overline{a}_{i,j} \mid i, j \in [n]\}$ and $A_t = \{c_i \mid i \in \lceil \log t \rceil\} \cup \{\overline{c}_i \mid i \in \lceil \log t \rceil\}$, and the item sets $I(n, n) = \{b_{i,j} \mid i, j \in [n]\} \cup \{\overline{b}_{i,j} \mid i, j \in [n]\}$ and $I_t = \{d_i \mid i \in \lceil \log t \rceil\} \cup \{\overline{d}_i \mid i \in \lceil \log t \rceil\}$. Both constructions are defined over the agent set $A = A(n, n) \cup A_t$ and the item set $I = I(n, n) \cup I_t$.

Intuitively, we use $A(m, n)$ and $I(m, n)$ in order to construct the preference profiles $P_1(m, n), P_2(m, n)$ and $P_3(D_i)$ for each $i \in \{0, \ldots, t-1\}$. The agents from $A_t$ “choose” a Yes-instance (if one exists) from $D_0, \ldots, D_{t-1}$; their goal is to enforce an assignment to be pareto optimal in (at most) one profile among $P_3(D_0), \ldots, P_3(D_{t-1})$.

The first layer is a composition of $P_1(n, n)$ with additional $2 \lceil \log t \rceil$ preference lists that belong to the agents from $A_t$ and is defined as follows:

- $a_{i,j} : b_{i,j} > \overline{b}_{i,j} \quad \forall i, j \in [n]$
- $\overline{a}_{i,j} : b_{i,j} > \overline{b}_{i,j} \quad \forall i, j \in [n]$
- $c_i : d_i > \overline{d}_i \quad \forall i \in \lceil \log t \rceil$
- $\overline{c}_i : d_i > \overline{d}_i \quad \forall i \in \lceil \log t \rceil$
The second layer consists of the preference profile $P_2(n, n)$ together with the same preferences of the agents from $A_t$ as in the first layer:

- $a_{n,j} : b_{n,j} > b_{n-1,j} > \overline{b_{n,j}} \forall j \in [n]$
- $\overline{a_{n,j}} : b_{n,j} > b_{n-1,j} > \overline{b_{n,j}} \forall j \in [n]$
- $a_{i,j} : b_{i-1,j} > b_{i,j} > b_{i+1,j} > b_{i,j} \forall i \in \{2, \ldots, n-1\}, j \in [n]$
- $\overline{a_{i,j}} : b_{i-1,j} > b_{i,j} > b_{i+1,j} > b_{i,j} \forall i \in \{2, \ldots, n-1\}, j \in [n]$
- $a_{1,j} : b_{1,j} > b_{2,j} > b_{1,j} \forall j \in [n]$
- $\overline{a_{1,j}} : b_{1,j} > b_{2,j} > b_{1,j} \forall j \in [n]$
- $c_i : d_i > \overline{d_i} \forall i \in \lceil\log t\rceil$
- $\overline{c_i} : d_i > \overline{d_i} \forall i \in \lceil\log t\rceil$

**Definition 12.** For each $j \in \lceil\log t\rceil$ and $b \in \{0, 1\}$, we define $d(b) = \begin{cases} d_j & b = 0 \\ \overline{d_j} & b = 1 \end{cases}$

We also denote $M^0_j = \{(c_j, d_j), (\overline{c_j}, \overline{d_j})\}$ and $M^1_j = \{(c_j, \overline{d_j}), (\overline{c_j}, d_j)\}$. Informally speaking, the sets $M^0_j$ and $M^1_j$ will correspond to the value of the $j$-th bit in the binary representation of $i \in \{0, \ldots, t-1\}$ such that $D_i$ is a Yes-instance. The sets $M^b_j$ that are contained in a pareto optimal assignment will “encode” the index of some Yes-instance (if one exists). For each $i \in \mathbb{N}, j \in \lceil\log i\rceil$, we define
i[j] as the j-th bit in the binary representation of i, i.e. $i = i[1]i[2]...i[\log t]$. We denote the set $M_i$ by $M_i = \bigcup_{j=1}^{\log t} M_i^{[j]}$. We first claim the following.

**Claim 10.** An assignment $p : A \rightarrow I \cup \{b_0\}$ is pareto optimal in the second two layers if and only if it is pareto optimal in both $P_1(n, n)$ and $P_2(n, n)$, and there exists exactly one $i \in \{0, ..., t - 1\}$ such that $M_i \subseteq p$.

*Proof. ($\Rightarrow$): Assume that $p$ is pareto optimal in both layers. Since $P_1(n, n)$ is contained in the first layer, $p$ is pareto optimal in $P_1(n, n)$. Similarly, $p$ is pareto optimal in $P_2(n, n)$ since $P_1(n, n)$ is contained in the second layer. Observe that for each $i \in [\log t]$, $d_i$ and $\overline{d_i}$ are both acceptable by $c_i$ and $\overline{c_i}$; and are the only items acceptable by them. Then, we have that \{p(c_i), p(\overline{c_i})\} = \{d_i, \overline{d_i}\} (else we get a self-loop), implying that for each $j \in [\log t]$, either $M_j^0 \subseteq p$ or $M_j^1 \subseteq p$. Let $i[j] = 0$ if $M_j^0 \subseteq p$ and $i[j] = 1$ if $M_j^1 \subseteq p$. Let $i \in \mathbb{N}$ be a number whose binary representation is $i = i[1]...i[\log t]$; then, we have that $M_i \subseteq p$. We prove that $i$ is unique. Towards a contradiction, suppose there exists $e \in \{0, ..., t - 1\}$ such that $e \neq i$ and $p$ satisfies that both $M_e \subseteq p$ and $M_e \subseteq p$. Since $e \neq i$, there exists a place in their binary representations where they differ, i.e. there exists $r \in [\log t]$ such that $e[r] \neq i[r]$. Thus, both $M_r^0 \subseteq p$ and $M_r^1 \subseteq p$, and this gives us a contradiction.

($\Leftarrow$): Assume that $p$ is pareto optimal in both $P_1(n, n)$ and $P_2(n, n)$, and there exists exactly one $i \in \{0, ..., t - 1\}$ such that $M_i \subseteq p$. First, we prove that $p$ is pareto optimal in the first layer: Observe that all the agents pairwise respect each other in the first layer; then, by Lemma 3, $p$ does not admit trading cycles in this layer. Moreover, by Claim 7, $p$ allocates all the items, hence it does not admit self loops. Thus, Proposition 1 implies that $p$ is pareto optimal in the first layer. Second, we prove that $p$ is pareto optimal in the second layer: Observe that the agents from $A_i$ respect each other in the second layer; then by Lemma 3, $p$ does not admit trading cycles among them. Furthermore, notice that the set of items acceptable by the agents from $A(n, n)$ is disjoint from the set of items acceptable by the agents from $A_i$, thus $p$ cannot admit trading cycles that consist of both agents from $A_i$ and agents from $A(n, n)$. Lastly, since $p$ is pareto optimal in $P_2(n, n)$, Proposition 1 implies that it does not admit trading cycles among agents from $A(n, n)$. Then, we have by Proposition 1 that $p$ is pareto optimal in the second layer.

We define the remaining $t$ layers separately for each construction.

The parameter $k = n + m + \alpha$. The cross-composition for this parameter constructs layer $3 + i$ as a composition of $P_3(D_i)$ with a unique combination of preference lists of length 1 that belong to agents from $A_i$. It is defined as follows:

- $a_{q, \text{ind}_{D_i}(q, 3)} : b_{D_i}(q, 3) > b_{D_i}(q, 2) > b_{D_i}(q, 3) \quad \forall q \in [m]$
- $a_{q, \text{ind}_{D_i}(q, 2)} : b_{D_i}(q, 2) > b_{D_i}(q, 1) > b_{D_i}(q, 2) \quad \forall q \in [m]$
- $a_{q, \text{ind}_{D_i}(q, 1)} : b_{D_i}(q, 1) > b_{D_i}(q, 3) > b_{D_i}(q, 1) \quad \forall q \in [m]$
The construction can be done in time that is polynomial in \( n + m + \alpha \). An assignment can be pareto optimal in at most one layer among layers 3, \ldots, 3 + (t - 1). If \( p \) is a 3-globally optimal assignment, then the layer 3 + i in which it is pareto optimal corresponds to a Yes-instance \( D_i \). This \( i \in \{0, \ldots, t - 1\} \) is encoded by the allocations of the agents from \( A_i \); \( P(A_i,i) \) is the preference profile in which \( c_j \) accepts only \( d(j,i[j]) \) and \( \tau_j \) accepts only \( d(j,i[j]) \), for each \( j \in [[\log t]] \).

\begin{itemize}
  \item \( a_{q,\text{ind}_{D_i}(q,r)} : b_{D_i}(q,r) > b_{D_i}(q,r) \) \( \forall q \in [m], r \in [3] \)
  \item \( a_{q,j} : b_{q,j} > b_{q,j} \) \( \forall q \in [m], j \in [n] \) such that \( x_j \) does not appear in the \( q \)-th clause in \( C_i \)
  \item \( a_{q,j} : b_{q,j} > b_{q,j} \) \( \forall q \in [m], j \in [n] \) such that \( x_j \) does not appear in the \( q \)-th clause in \( C_i \)
  \item \( c_j : d(j,i[j]) \) \( \forall j \in [[\log t]] \)
  \item \( \tau_j : \overline{d}(j,i[j]) \) \( \forall j \in [[\log t]] \)
\end{itemize}

An illustration of the last \( t \) layers in shown in Figure 9. Intuitively, the goal of the agents from \( A_i \) is to enforce each assignment to be pareto optimal in at most one layer among 3, \ldots, 3 + (t - 1). We finally set \( \alpha = 3 \). Note that the construction can be done in time that is polynomial in \( \Sigma_{i=1}^{t-1}|D_i| \). Before we prove the correctness of the construction, let us prove the following:

**Claim 11.** Let \( p : A \rightarrow I \cup \{b_0\} \) be an assignment and \( i \in \{0, \ldots, t - 1\} \). Then \( p \) is pareto optimal in layer 3 + i if and only if it is pareto optimal in \( P_3(D_i) \) and \( M_i \subseteq p \).

**Proof.** \((\Rightarrow)\): Assume that \( p \) is pareto optimal in layer 3 + i. First, since \( P_3(D_i) \) is contained in the preference profile of layer 3 + i, we infer that \( p \) is pareto optimal in \( P_3(D_i) \). Second, observe that for each \( i \in [[\log t]] \), \( c_j \) only accepts \( d(j,i[j]) \) and \( \tau_j \) only accepts \( \overline{d}(j,i[j]) \). Furthermore, \( d(j,i[j]) \) and \( \overline{d}(j,i[j]) \) are only acceptable by these agents. This enforces \( p \) to satisfy \( p(c_j) = d(j,i[j]) \) and \( p(\tau_j) = \overline{d}(j,i[j]) \) otherwise, \( p \) would admit self loops. Hence, \( p \) satisfies \( M_i \subseteq p \).

\((\Leftarrow)\): Assume that \( p \) is pareto optimal in \( P_3(D_i) \) and that \( M_i \subseteq p \). Notice that \( p \) allocates all the items, thus it cannot admit self loops in layer 3 + i. In addition, since each agent from \( A_i \) only accepts a single item, \( p \) cannot admit trading cycles among agents from \( A_i \). Moreover, since \( P_3(D_i) \) is contained in the preference
profile in layer $3 + i$, and $p$ is pareto optimal in $P_3(D_i)$, by Proposition 1 we have that $p$ does not admit trading cycles among agents from $A(n, n)$. We have also that $p$ does not admit trading cycles that contain both agents from $A_i$ and agents from $A(n, n)$ since their sets of acceptable items are disjoint. Then we infer that $p$ is pareto optimal in layer $3 + i$. \hfill $\square$

**Claim 12.** Let $p : A \to I \cup \{b_0\}$ be an assignment. Then there exists at most one $i \in \{0, \ldots, t - 1\}$ such that $p$ is pareto optimal in layer $3 + i$.

*Proof.* Towards a contradiction, suppose there exist $i_1, i_2 \in \{0, \ldots, t - 1\}$ such that $i_1 \neq i_2$ and $p$ is pareto optimal in both layers $3 + i_1$ and $3 + i_2$. By Claim 11, both $M_{i_1} \subseteq p$ and $M_{i_2} \subseteq p$, a contradiction to Claim 10. \hfill $\square$

We now prove that there exists $i \in \{0, \ldots, t - 1\}$ such that $D_i$ is a *Yes*-instance of 3-SAT if and only if there exists a 3-globally optimal assignment for the constructed instance.

($\Rightarrow$): Assume there exists $i \in \{0, \ldots, t - 1\}$ such that $D_i$ is a *Yes*-instance. By Lemma 4, there exists an assignment $p$ that is pareto optimal in the profiles $P_1(n, n), P_2(n, n)$ and $P_3(D_i)$. We extend $p$ by $p \leftarrow p \cup M_i$. Since $p$ allocates all the items from $I$, it is pareto optimal in the first two layers by Claim 10. By Claim 11, it is pareto optimal in layer $3 + i$. Thus, $p$ is 3-globally optimal for the constructed instance.

($\Leftarrow$): Suppose there exists a 3-globally optimal assignment for the constructed instance. By Claim 12, $p$ is pareto optimal in at most one among layers $3, \ldots, 3 + (t - 1)$, say, layer $3 + i$ where $i \in \{0, \ldots, t - 1\}$. By Claim 11, $p$ is pareto optimal in $P_3(D_i)$ and by Claim 10, it is pareto optimal in both $P_1(n, n)$ and $P_2(n, n)$. Thus we have by Lemma 4 that $D_i$ is a *Yes*-instance.

Since the constructed instance satisfies $n + m + \alpha = \mathcal{O}(n^2 + \log t)$, Proposition 7 implies that $\alpha$-**GLOBALLY OPTIMAL ASSIGNMENT** does not admit a polynomial kernel with respect to the parameter $k = n + m + \alpha$, unless $\text{NP} \subseteq \text{coNP/poly}$.

**The parameter** $k = n + m + (\ell - \alpha)$. Similarly to the previous cross-composition, the goal of the agents in $A_i$ here is to encode some $i \in \{0, \ldots, t - 1\}$ such that $D_i$ is a *Yes*-instance (if one exists). This is done by enforcing every $\ell$-pareto optimal assignment to be pareto optimal in some profile $P_3(D_i)$ and preventing the existence of trading cycles in all layers $3 + j$ where $j \neq i$. Let $i \in \{0, \ldots, t - 1\}$, we define layer $3 + i$ as follows:

- $a_{q, \text{ind}_{D_i}(q, 3)} : b_{D_i}(q, 3) > b_{D_i}(q, 2) > b_{D_i}(q, 3) \quad \forall q \in [m]$
- $a_{q, \text{ind}_{D_i}(q, 2)} : b_{D_i}(q, 2) > b_{D_i}(q, 1) > b_{D_i}(q, 2) \quad \forall q \in [m]$
- $a_{q, \text{ind}_{D_i}(q, 1)} : b_{D_i}(q, 1) > d([\log t], i([\log t])) > b_{D_i}(q, 1) \quad \forall q \in [m]$
- $a_{q, \text{ind}_{D_i}(q, r)} : b_{D_i}(q, r) > b_{D_i}(q, r) \quad \forall q \in [m], r \in [3]$
- $a_{q, j} : b_{q, j} > b_{q, j} \quad \forall q \in [m], j \in [n]$ such that $x_j$ does not appear in the $q$-th clause in $C_i$
Parameterized Analysis of Assignment Under Multiple Preferences

Layer 3

\[ P_3(D_l) \]

\[ Q(A_t,1) \]

Layer 3+i

\[ P_3(D_l) \]

\[ Q(A_t,i) \]

Layer 3+(t-1)

\[ P_3(D_l) \]

\[ Q(A_t,t) \]

Fig. 10. The last \( t \) layers constructed for the parameter \( n+m+(\ell-\alpha) \). \( Q(A_t,i) \) consists of the preference lists of the agents from \( A_t \) in layer \( 3+i \), for each \( i \in \{0, \ldots, t-1\} \).

All the possible cycles in layers \( 3, \ldots, 3+(t-1) \) can be decomposed into a path with 3 items in the trading graph of \( P_3(D_l) \), connected to a path with \( \lceil \log t \rceil \) items in the trading graph of \( Q(A_t,i) \), and an edge connecting the end point of the second path to the start point of the first path. For an \( \ell \)-globally optimal assignment, the second path will be cut in most layers, then no cycles will exist. There will exist exactly one layer \( 3+i \) for which the trading graph of \( Q(A_t,i) \) contains such path. This implies that the trading graph of \( P_3(D_l) \) must not contain paths with 3 items, then \( D_l \) is a Yes-instance.

- \( a_{q,j} : b_{q,j} > b_{q,j} \forall q \in [m], j \in [n] \) such that \( x_j \) does not appear in the \( q \)-th clause in \( C_i \)
- \( c_2 : d(j,i[j]) > d(j-1,j[j-1]) > d(j,i[j]) \forall j \in \{2, \ldots, \lceil \log t \rceil \} \)
- \( c_1 : d(1,i[1]) > \{b_{D,(q,3)} \mid \forall q \in [m]\} > d(1,i[1]) \)
- \( r_j : d(j,i[j]) > d(j,i[j]) \forall j \in \lceil \log t \rceil \)

The construction is illustrated in Figure 10. We now claim the following.

Claim 13. Let \( p : A \rightarrow I \cup \{b_{\emptyset}\} \) be an assignment such that \( p \) is pareto optimal in the first two layers and let \( i \in \{0, \ldots, t-1\} \). Then \( p \) is pareto optimal in layer \( 3+i \) if and only if: (1) \( p \) allocates all the items in \( I \), and (2) \( M_i \not\subseteq p \) or \( p \) is pareto optimal in \( P_3(D_l) \).
Proof. (⇒): Assume that \( p \) is pareto optimal in layer \( 3 + i \). By Claim 10, \( p \) is pareto optimal in both \( P_1(n,n) \) and \( P_2(n,n) \) and there exists exactly one \( j \in \{0, \ldots, t - 1\} \) such that \( M_j \subseteq p \). By the construction of the first layer, \( p \) must allocate all the items since otherwise it would admit self loops. Notice that the only possible trading cycle in layer \( 3 + i \), for any \( i \in \{0, \ldots, t - 1\} \), has the form: \( (a_{q,\text{ind}_D(q,3)}, b_{D_i}(q,3), a_{q,\text{ind}_D(q,2)}, b_{D_i}(q,2), a_{q,\text{ind}_D(q,1)}, b_{D_i}(q,1), c_{[\log t]}, d([\log t], i[\log t]), \ldots, c_1, d(1, i[1])) \) and it exists only when \( p \) is not pareto optimal in \( P_3(D_i) \) and \( M_i \not\subseteq p \).

(⇐): Suppose that \( p \) satisfies that \( M_i \not\subseteq p \) or \( p \) is pareto optimal in \( P_3(D_i) \). By the observation in the previous direction, \( p \) does not admit trading cycles in layer \( 3 + i \). Thus, \( p \) is pareto optimal in layer \( 3 + i \).

We prove that there exists \( i \in \{0, \ldots, t - 1\} \) such that \( D_i \) is a Yes-instance if and only if there exists an \( \ell \)-globally optimal assignment for the constructed instance.

(⇒): Suppose there exists \( i \in \{0, \ldots, t - 1\} \) such that \( D_i \) is a Yes-instance. We extend \( p \) by \( p \leftarrow p \cup M_i \). By Lemma 4, \( p \) is pareto optimal in \( P_1(n,n), P_2(n,n) \) and \( P_3(D_i) \). First, since all the items in \( I_i \) are allocated, by Claims 11 and 12 we have that \( p \) is pareto optimal in the first two layers. Second, by Claim 13 we have that \( p \) is pareto optimal in layer \( 3 + i \). Third, let \( j \in \{0, \ldots, t - 1\} \setminus \{i\} \). By the construction of \( p \) we have that \( M_j \not\subseteq p \), then by Claim 13 \( p \) is pareto optimal in layer \( 3 + j \). Then we conclude that \( p \) is \( \ell \)-globally optimal for the constructed instance.

(⇐): Suppose that there exists an \( \ell \)-globally optimal assignment for the constructed instance \( p \). By Claims 11 and 12 \( p \) is pareto optimal in both \( P_1(n,n) \) and \( P_2(n,n) \) and there exists \( i \in \{0, \ldots, t - 1\} \) such that \( M_i \subseteq p \). Claim 13 implies that \( p \) is pareto optimal in \( P_3(D_i) \). Thus, by Lemma 4, \( D_i \) is a Yes-instance.

We show the the same result holds also for the parameter \( \#\text{items} + \ell \).

**Theorem 12.** There does not exist a polynomial kernel for \( \alpha \)-Globally Optimal Assignment with respect to \( k = \#\text{items} + \ell \), unless \( \text{NP} \subseteq \text{coNP/poly} \).

Proof. We present a cross-composition from 3-SAT to \( \alpha \)-Globally Optimal Assignment. Given instances of 3-SAT \( D_0 = (X_0,C_0), \ldots, D_{t-1} = (X_{t-1},C_{t-1}) \) of the same size \( n \in \mathbb{N} \) for some \( t \in \mathbb{N} \), we first modify each instance \( D_i \) to have \( X_i = \{x_1, \ldots, x_n\} \) and \( |C_i| = n \) using Observation 4. We define an agent set \( A_i(n,n) \) of \( n^2 \) agents for each instance \( D_i \) by \( A_i(n,n) = \{a^r_{i,j}, a^i_{r,j} | r,j \in [n]\} \) and we set \( A = \bigcup_{i=0}^{t-1} A_i(n,n) \). We also create the item set \( I = I(n,n) \). The constructed instance is defined over \( A \) and \( I \), and it consists of \( 2[\log t] + 2 \) layers. Notice that we have a total number of \( 2tn^2 \) agents and \( 2n^2 \) items, then in every assignment for the constructed instance, there will exist agents which get no items. Intuitively, the goal of the first \( 2[\log t] \) layers is to enforce every \( \alpha \)-globally optimal assignment to allocate the items in \( I \) only to agents that correspond to a Yes-instance (if one exists). They are constructed as compositions the profile \( P_1(n,n) \) over the agent set \( A_i(n,n) \) and the item set \( I(n,n) \) for
each $i \in \{0, \ldots, t - 1\}$ (namely, replacing each $a_{q,j}$ with $a_{q,j}^i$ in $P_i(n, n)$). Layer $2[\log t] + 1$ is constructed as a composition of the profile $P_2(n, n)$ over $A_i(n, n)$ and $I(n, n)$ for each $i \in \{0, \ldots, t - 1\}$, and the last layer is a composition of the profiles $P_3(D_j)$ over $A_i(n, n)$ and $I(n, n)$ for each $i \in \{0, \ldots, t - 1\}$. We first construct the first $[\log t]$ layers. Informally speaking, for $i \in \{[\log t]\}$, the preference profile in layer $i$ requires an assignment to allocate $b_q$ to any agent whose corresponding instance is $D_j$ such that the $i$-th bit in the binary representation of $j$ is 0. Layer $i$ is formally defined as follows:

- $a_{q,j}^i : b_{q,j} > b_{q,j}^i \forall q, j \in [n], r \in \{0, \ldots, t - 1\}$ such that $r[i] = 0$
- $a_{q,j}^i : b_{q,j} > b_{q,j}^i \forall q, j \in [n], r \in \{0, \ldots, t - 1\}$ such that $r[i] = 0$
- $a_{q,j}^i : \emptyset \forall q, j \in [n], r \in \{0, \ldots, t - 1\}$ such that $r[i] = 1$
- $a_{q,j}^i : \emptyset \forall q, j \in [n], r \in \{0, \ldots, t - 1\}$ such that $r[i] = 1$

We define the next $[\log t]$ layers. Intuitively, layer $[\log t] + i$ is different from layer $i$ such that it requires an assignment to allocate $b_q$ to any agent whose corresponding instance is $D_j$ such that the $i$-th bit in the binary representation of $j$ is 1 (instead of 0). For each $i \in \{[\log t]\}$, layer $[\log t] + i$ is defined as follows:

- $a_{q,j}^i : b_{q,j} > b_{q,j}^i \forall q, j \in [n], r \in \{0, \ldots, t - 1\}$ such that $r[i] = 1$
- $a_{q,j}^i : b_{q,j} > b_{q,j}^i \forall q, j \in [n], r \in \{0, \ldots, t - 1\}$ such that $r[i] = 1$
- $a_{q,j}^i : \emptyset \forall q, j \in [n], r \in \{0, \ldots, t - 1\}$ such that $r[i] = 0$
- $a_{q,j}^i : \emptyset \forall q, j \in [n], r \in \{0, \ldots, t - 1\}$ such that $r[i] = 0$

Layer $2[\log t] + 1$ is a composition of the profile $P_2(n, n)$ over the agent set $A_i(n, n)$ and the item set $I(n, n)$ for each $i \in [t]$. The layers are defined as follows:

- $a_{n,j}^i : b_{n,j} > b_{n-1,j} > b_{n,j}^i \forall i \in \{0, \ldots, t - 1\}, j \in [n]$
- $a_{n,j}^i : b_{n,j} > b_{n-1,j} > b_{n,j}^i \forall i \in \{0, \ldots, t - 1\}, j \in [n]$
- $a_{q,j}^i : b_{q-1,j} > b_{q,j} > b_{q+1,j} > b_{q,j}^i \forall i \in \{0, \ldots, t - 1\}, q \in \{2, \ldots, n - 1\}, j \in [n]$
- $a_{q,j}^i : b_{q-1,j} > b_{q,j} > b_{q+1,j} > b_{q,j}^i \forall i \in \{0, \ldots, t - 1\}, q \in \{2, \ldots, n - 1\}, j \in [n]$
- $a_{1,j}^i : b_{1,j} > b_{2,j} > b_{1,j} \forall i \in \{0, \ldots, t - 1\}, j \in [n]$
- $a_{1,j}^i : b_{1,j} > b_{2,j} > b_{1,j} \forall i \in \{0, \ldots, t - 1\}, j \in [n]$
Layer 2 \([\log t] + 2\) is a composition of the profiles \(P_3(D_i)\) over \(A_i(n, n)\) and \(I(n, n)\) for each \(i \in \{0, \ldots, t-1\}\).

\[
\begin{align*}
&\bullet a_{j,\text{ind},D_i,(j,3)}: b_{D_i}(j, 3) > b_{D_i}(j, 2) > b_{D_i}(j, 3) \quad \forall i \in \{0, \ldots, t-1\}, j \in [n] \\
&\bullet a_{j,\text{ind},D_i,(j,2)}: b_{D_i}(j, 2) > b_{D_i}(j, 1) > b_{D_i}(j, 2) \quad \forall i \in \{0, \ldots, t-1\}, j \in [n] \\
&\bullet a_{j,\text{ind},D_i,(j,1)}: b_{D_i}(j, 1) > b_{D_i}(j, 3) > b_{D_i}(j, 1) \quad \forall i \in \{0, \ldots, t-1\}, j \in [n] \\
&\bullet a_{j,\text{ind},D_i,(j,r)}: b_{D_i}(j, r) > b_{D_i}(j, r) \quad \forall i \in \{0, \ldots, t-1\}, j \in [n], r \in [3] \\
&\bullet a_{q,r}: b_{q,r} > \overline{b}_{q,r} \quad \forall i \in \{0, \ldots, t-1\}, q, r \in [n] \text{ such that } x_r \text{ does not appear in the } q\text{-th clause in } C_i \\
&\bullet a_{q,r}: b_{q,r} > \overline{b}_{q,r} \quad \forall i \in \{0, \ldots, t-1\}, q, r \in [n] \text{ such that } x_r \text{ does not appear in the } q\text{-th clause in } C_i.
\end{align*}
\]

We finally set \(\alpha = [\log t] + 2\). We claim the following:

**Claim 14.** Let \(p: A \to I \cup \{b_0\}\) be an assignment for the constructed instance and let \(i \in [\log t]\). Then \(p\) satisfies the following:

- \(p\) is pareto optimal in layer \(i\) if and only if it allocates all the items in \(I\) to agents from the set \(\{a_{q,j}, \overline{a}_{q,j} | q, j \in [n], r \in \{0, \ldots, t-1\} \text{ such that } r[i] = 0\}\).
- \(p\) is pareto optimal in layer \([\log t] + i\) if and only if it allocates all the items in \(I\) to agents from the set \(\{a_{q,j}, \overline{a}_{q,j} | q, j \in [n], r \in \{0, \ldots, t-1\} \text{ such that } r[i] = 1\}\).

**Proof.** We provide a proof for the first condition (the proof for the second is symmetric).

(\(\Rightarrow\)): Assume that \(p\) is pareto optimal in layer \(i\). Since the preference lists of agents \(a_{q,j}\) and \(\overline{a}_{q,j}\) for all \(q, j \in [n]\) and \(r \in \{0, \ldots, t-1\}\) such that \(r[i] = 1\) are empty, \(p\) allocates \(b_0\) to all such agents, and therefore allocates items only to agents from \(\{a_{q,j}, \overline{a}_{q,j} | q, j \in [n], r \in \{0, \ldots, t-1\} \text{ such that } r[i] = 0\}\). In addition, observe that each item in \(I\) is acceptable by at least two agents from the set \(\{a_{q,j}, \overline{a}_{q,j} | q, j \in [n], r \in \{0, \ldots, t-1\} \text{ such that } r[i] = 0\}\). By Proposition 1, \(p\) does not admit self loops. Thus we have that all the items in \(I\) are allocated.

(\(\Leftarrow\)): Suppose that \(p\) allocates all the items in \(I\) to agents from the set \(\{a_{q,j}, \overline{a}_{q,j} | q, j \in [n], r \in \{0, \ldots, t-1\} \text{ such that } r[i] = 0\}\). First, since \(p\) allocates all the items, it does not admit self loops in layer \(i\). Second, observe that all the agents pairwise respect each other in this layer, thus by Lemma 1, \(p\) does not admit trading cycle. By Proposition 1, \(p\) is pareto optimal in layer \(i\).

**Claim 15.** Let \(p: A \to I \cup \{b_0\}\) be an assignment for the constructed instance. Then there does not exist \(i \in [\log t]\) such that \(p\) is pareto optimal in both layers \(i\) and \([\log t] + i\).
Claim 16. Let \( p : A \to I \cup \{b_0\} \) be an assignment for the constructed instance. Then \( p \) is pareto optimal in at most \( \lceil \log t \rceil \) layers among the first \( 2\lceil \log t \rceil \) layers.

Proof. Claim [15] implies that for each \( i \in [\log t] \), \( p \) is pareto optimal in at most one among layers \( i \) and \( \lceil \log t \rceil + i \). Implying that \( p \) is pareto optimal in at most \( \lceil \log t \rceil \) layers among the first \( 2\lceil \log t \rceil \) layers.

Claim 17. Let \( p : A \to I \cup \{b_0\} \) be an \( \alpha \)-globally optimal assignment for the constructed instance. Then there exists \( i \in \{0, \ldots, t-1\} \) that \( p \) encodes in the first \( 2\lceil \log t \rceil \) layers.

Proof. Since \( \alpha = \lceil \log t \rceil + 2 \), Claim [16] implies that \( p \) is pareto optimal in exactly \( \lceil \log t \rceil \) layers among the first \( 2\lceil \log t \rceil \) layers. By Claim [15] for each \( i \in [\log t] \), \( p \) is pareto optimal in either layer \( j \) or layer \( \lceil \log t \rceil + j \). Denote \( i[j] = 0 \) if \( p \) is pareto optimal in layer \( j \) and \( i[j] = 1 \) if \( p \) is pareto optimal in layer \( \lceil \log t \rceil + j \). Observe that \( p \) encodes \( i = i[1], i[2], \ldots, i[\lceil \log t \rceil] \) in the first \( 2\lceil \log t \rceil \) layers.

Claim 18. Let \( p : A \to I \cup \{b_0\} \) be an assignment for the constructed instance. Then there exists \( i \in \{0, \ldots, t-1\} \) such that \( p \) encodes in the first \( 2\lceil \log t \rceil \) layers if and only if \( p \) is pareto optimal in \( P_1(n, n) \) over the agent set \( A_1(n, n) \) and the item set \( I(n, n) \).

Proof. (\( \Rightarrow \)): Assume that there exists \( i \in \{0, \ldots, t-1\} \) that \( p \) encodes in the first \( 2\lceil \log t \rceil \) layers. Observe that for each \( j \in [\lceil \log t \rceil] \), if \( i[j] = 0 \) then \( p \) is pareto optimal in layer \( j \), and if \( i[j] = 1 \) then \( p \) is pareto optimal in layer \( \lceil \log t \rceil + j \). By Claim [14] \( p \) allocates all the items in \( I \) to agents from \( \bigcap_{j=1}^{[\log t]} \{ a_{q,r}^s, \overline{a_{q,r}}^s \mid q, s \in [n], r \in \{0, \ldots, t-1\} \} \) such that \( r[j] = i[j] \) \( \in A_1(n, n) \). Since the agents in \( A_1(n, n) \) respect each other in \( P_1(n, n) \) over \( A_1(n, n) \) and \( I(n, n) \), and all the items are allocated to \( A_1(n, n) \), we have by Lemma [3] and Proposition [4] that \( p \) is pareto optimal in \( P_1(n, n) \) over \( A_1(n, n) \) and \( I(n, n) \).

(\( \Leftarrow \)): Suppose that \( p \) is pareto optimal in \( P_1(n, n) \) over \( A_1(n, n) \) and \( I(n, n) \). We claim that \( p \) encodes \( i \) in the first \( 2\lceil \log t \rceil \) layers: By Claim [14] for each \( j \in [\lceil \log t \rceil] \), if \( i[j] = 0 \) then \( p \) is pareto optimal in layer \( j \), and if \( i[j] = 1 \) then \( p \) is pareto optimal in layer \( \lceil \log t \rceil + j \). Then we have that \( p \) encodes \( i \) in the first \( 2\lceil \log t \rceil \) layers. □
Claim 19. Let \( p : A \to I \cup \{b_0\} \) be an assignment such that there exists \( i \in \{0, \ldots, t-1\} \) such that \( p \) encodes \( i \) in the first \( 2[\log t] \) layers. Then \( p \) is pareto optimal in layer \( 2[\log t] + 1 \) if and only if it is pareto optimal in \( P_2(n, n) \) over \( A_i(n, n) \) and \( I(n, n) \).

Proof. (\( \Rightarrow \)): Assume that \( p \) is pareto optimal in layer \( 2[\log t] + 1 \). Since \( P_2(n, n) \) over \( A_i(n, n) \) and \( I(n, n) \) is contained in the preference profile of layer \( 2[\log t]+1 \), \( p \) must be pareto optimal in this sub-profile.

(\( \Leftarrow \)): Assume that \( p \) is pareto optimal in \( P_2(n, n) \) over \( A_i(n, n) \) and \( I(n, n) \). By the construction of \( P_2(n, n) \) over \( A_i(n, n) \) and \( I(n, n) \), \( p \) must allocate all the items to agents from \( A_i(n, n) \), as otherwise, it would admit self loops. Observe that agents from \( A \setminus A_i(n, n) \) do not admit trading cycles since they are assigned to \( b_0 \), and they also do not admit self loops since all their acceptable items are already allocated by \( p \).

Claim 20. Let \( p : A \to I \cup \{b_0\} \) be an assignment such that there exists \( i \in \{0, \ldots, t-1\} \) such that \( p \) encodes \( i \) in the first \( 2[\log t] \) layers. Then \( p \) is pareto optimal in layer \( 2[\log t] + 2 \) if and only if it is pareto optimal in \( P_2(D_i) \) over \( A_i(n, n) \) and \( I(n, n) \).

Proof. (\( \Rightarrow \)): Assume that \( p \) is pareto optimal in layer \( 2[\log t] + 2 \). Since \( P_2(n, n) \) over \( A_i(n, n) \) and \( I(n, n) \) is contained in the preference profile of layer \( 2[\log t]+1 \), \( p \) must be pareto optimal in this sub-profile.

(\( \Leftarrow \)): Assume that \( p \) is pareto optimal in \( P_2(n, n) \) over \( A_i(n, n) \) and \( I(n, n) \). By the construction of \( P_2(n, n) \) over \( A_i(n, n) \) and \( I(n, n) \), \( p \) must allocate all the items to agents from \( A_i(n, n) \), as otherwise, it would admit self loops. Observe that agents in \( A \setminus A_i(n, n) \) do not admit trading cycles since they are assigned to \( b_0 \), and they also do not admit self loops since all their acceptable items are already allocated by \( p \).

We now prove the correctness of the construction. Namely, there exists \( i \in \{0, \ldots, t-1\} \) such that \( D_i \) is a Yes-instance of 3-SAT if and only if there exists an \( \alpha \)-globally optimal for the constructed instance.

(\( \Rightarrow \)): Suppose there exists \( i \in \{0, \ldots, t-1\} \) such that \( D_i \) is a Yes-instance. By Lemma 4, there exists an assignment \( q : A(n, n) \to I(n, n) \cup \{b_0\} \) such that \( q \) is pareto optimal in \( P_3(n, n) \) over \( A_i(n, n) \) and \( I(n, n) \). We define an assignment \( p : A \to I \cup \{b_0\} \) by \( p(a^i_{r,j}) = q(a^i_{r,j}) \), \( p(a^{\neg i}_{r,j}) = q(n_{r,j}) \) for each \( j, r \in [n] \), and \( p(a^j_{r,0}) = b_0 \), \( p(a^{\neg j}_{r,0}) = b_0 \) for each \( j, r \in [n] \), \( y \in \{0, \ldots, t-1\} \setminus \{i\} \). By Claims 14 and 16, we have that \( p \) is pareto optimal in exactly \( [\log t] \) layers among the first \( 2[\log t] \) layers and that it encodes \( i \). By Claims 19 and 20, \( p \) is pareto optimal in both layers \( 2[\log t]+1 \) and \( 2[\log t]+2 \). Thus, \( p \) is an \( \alpha \)-globally optimal assignment for the constructed instance.

(\( \Leftarrow \)): Assume that there exists \( p : A \to I \cup \{b_0\} \) that is \( \alpha \)-globally optimal for the constructed instance (see Figure 11). By Claims 16 and 17, \( p \) is pareto optimal in exactly \( [\log t] \) layers among the first \( 2[\log t] \) layers, and there exists \( i \in \{0, \ldots, t-1\} \) such that \( p \) encodes \( i \) in the first \( 2[\log t] \) layers. By Claim 18
If \( p \) is \((\lceil \log t \rceil + 2)\)-globally optimal for the constructed instance, then the \( \lceil \log t \rceil \) layers in which it is pareto optimal among the first \( 2\lceil \log t \rceil \) layers encode \( i \in \{0, \ldots, t-1\} \) such that: (1) \( p \) allocates all the items from \( I(n, n) \) to all the agents from \( A_i(n, n) \), and (2) \( p \) is pareto optimal in \( P_1(n, n) \) over \( A_i(n, n) \) and \( I(n, n) \). We have that \( p \) must be pareto optimal in both layers \( 2\lceil \log t \rceil + 1 \) and \( 2\lceil \log t \rceil + 2 \), implying that it is pareto optimal in both \( P_2(n, n) \) and \( P_3(D_i) \) over \( A_i(n, n) \) and \( I(n, n) \).

\( p \) is pareto optimal in \( P_1(n, n) \) over \( A_i(n, n) \) and \( I(n, n) \). Since \( p \) is \( \alpha \)-globally optimal, it is pareto optimal in layers \( 2\lceil \log t \rceil + 1 \) and \( 2\lceil \log t \rceil + 2 \). Claims 19 and 20 imply that \( p \) is pareto optimal in both \( P_2(n, n) \) and \( P_3(D_i) \) over \( A_i(n, n) \) and \( I(n, n) \). Hence, by Lemma 4 we have that \( D_i \) is a \textbf{Yes}-instance.

Since the constructed instance satisfies \( m + \ell = \mathcal{O}(n^2 + \log(t)) \), we have by Proposition 4 that \( \alpha \)-GLOBALLY OPTIMAL ASSIGNMENT does not admit a polynomial kernel with respect to the parameter \( k = m + \ell \), unless \( \text{NP} \subseteq \text{coNP/poly} \).

\[\Box\]

6 Conclusion and Future Research

In this paper, we introduced a new variant of the ASSIGNMENT problem where each agent is equipped with multiple incomplete preference lists, and we defined a corresponding notion of global optimality, that naturally extends pareto optimality. We considered several natural parameters, and presented a compre-
hensive picture of the parameterized complexity of the problem with respect to them.

The results show that the problem of finding an $\alpha$-globally optimal assignment is, in general, computationally hard, but that it admits more positive results when the parameter depends on $n = \#\text{agents}$ (and $\alpha$ or $\ell$) than when it depends on $m = \#\text{items}$ (and $\alpha$ or $\ell$). We proved that the problem admits an XP algorithm with respect to $m$, but is unlikely to admit one with respect to $\ell + d$ and $\ell + (m - d)$. We provided an $O^*(n!)$-time algorithm and an exponential kernel with respect to $m + \ell$. Both results showed that the problem is FPT with respect to these parameters. In addition, we proved that $O^*(k!)$ is essentially a tight lower bound on the running time under ETH for even larger parameters than $n$ such as $k = n + m + \alpha$ and $k = n + m + (\ell - \alpha)$. Moreover, we proved that the problem admits a polynomial kernel with respect to $m + \ell$, but is unlikely to admit one with respect to all the other parameters that we considered. We also proved that the problem is $W[1]$-hard with respect to $m + \alpha$ and $m + (\ell - \alpha)$.

However, two questions are still open:

1. Is it possible to obtain a (not polynomial) better kernel for $m + \ell$ with size substantially smaller than $O^*((m!)^\ell + 1)$?

2. Is it possible to obtain an algorithm with a better running time than $O^*(k!)$ for $k = n + m + \ell$? It can be shown using a reduction from 3-SAT (similar to the reduction from 3-SAT in Section 4) that the problem cannot be solved in time $2^{o(k)}$ under ETH. However, can a $2^{O(k)}$-algorithm be achieved?

Continuing our research, it might be interesting to study “weaker” definitions of optimality, for example: finding an assignment such that every group of $k$ agents has some $\alpha$ layers where they (1) do not admit trading cycles; (2) are not parts of larger trading cycles; or (3) do not admit the same trading cycle. Verification variants of these problems can also be suggested, i.e. given an assignment $p$, check whether it is optimal.

Another direction is to study the particular case where the preferences of the agents are complete since it may provide more positive algorithmic results under some parameterizations. In addition, notice that a solution to $\alpha$-GLOBALLY OPTIMAL ASSIGNMENT can be seen as an approximation to the “optimal” solution in which an assignment is pareto optimal in a maximum number of layers (this is similar to the VERTEX COVER problem, where the parameter $k$ is somewhat an “approximation” to the size of the minimum vertex cover). In this approach, we can define the problem as an approximation problem and study it from the perspective of parameterized approximation.

In this paper, we considered the basic “unweighed” model of the problem (since this is the first study of this kind). Another direction is to consider a weighted version in which some criteria (layers) may have higher importance than others. A straightforward way to model this is by having several copies of layers. However, if weights are high and varied, this might lead to inefficiency.
References

1. Abdulkadirog, A.: House allocation with existing tenants 1 (1999)
2. Abdulkadiroglu, A., Sonmez, T.: Random Serial Dictatorship and the Core from Random Endowments in House Allocation Problems. Econometrica 66(3), 689–702 (May 1998). https://ideas.repec.org/a/ecm/emetrp/v66y1998i3p689-702.html
3. Abraham, D.J., Cechlárová, K., Manlove, D.F., Mehlhorn, K.: Pareto optimality in house allocation problems. In: Proceedings of the 15th International Conference on Algorithms and Computation, p. 3–15. ISAAC’04, Springer-Verlag, Berlin, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30551-4_3
4. Alderfer, C.P., McCord, C.G.: Personal and situational factors in the recruitment interview. Journal of Applied Psychology 54(4), 377 (1970)
5. Aziz, H., Biro, P., de Haan, R., Rastegari, B.: Pareto optimal allocation under compact uncertain preferences. In: Thirty Third AAAI Conference on Artificial Intelligence (01/02/19) (October 2018). https://eprints.soton.ac.uk/425734/
6. Aziz, H., de Haan, R., Rastegari, B.: Pareto optimal allocation under uncertain preferences. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, pp. 77–83 (2017). https://doi.org/10.24963/ijcai.2017/12
7. Bodlaender, H.L., Downey, R.G., Fellows, M.R., Hermelin, D.: On problems without polynomial kernels. J. Comput. Syst. Sci. 75(8), 423–434 (2009). https://doi.org/10.1016/j.jcss.2009.04.001
8. Bodlaender, H.L., Jansen, B.M.P., Kratsch, S.: Kernelization lower bounds by cross-composition. SIAM J. Discrete Math. 28(1), 277–305 (2014). https://doi.org/10.1137/120880240
9. Bogomolnaia, A., Moulin, H.: A new solution to the random assignment problem. J. Economic Theory 100, 295–328 (2001)
10. Browne, A.: Lives ruined as nhs leaks patients’ notes. The Observer, June 25th (2000)
11. Calabro, C., Impagliazzo, R., Paturi, R.: The complexity of satisfiability of small depth circuits. In: Parameterized and Exact Computation, 4th International Workshop, IWPEC 2009, Copenhagen, Denmark, September 10-11, 2009, Revised Selected Papers. pp. 75–85 (2009). https://doi.org/10.1007/978-3-642-11269-0_6
12. Chen, J., Niedermeier, R., Skowron, P.: Stable marriage with multi-modal preferences. In: Proceedings of the 2018 ACM Conference on Economics and Computation. p. 269–286. EC ’18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3219166.3219168
13. Chernichovsky, D., Kfir, R.: The state of the acute care hospitalization system in Israel
14. Clinedinst, M.: State of college admission (2019)
15. Cools, M., Moons, E., Wets, G.: Assessing the impact of weather on traffic intensity. Weather, Climate, and Society 2(1), 60–68 (2010)
16. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer (2015).
17. Downey, R.G., Fellows, M.R.: Fundamentals of Parameterized Complexity. Texts in Computer Science, Springer (2013). https://doi.org/10.1007/978-3-540-21275-3

18. Faudzi, S., Abdul-Rahman, S., Abd Rahman, R.: An assignment problem and its application in education domain: A review and potential path. Advances in Operations Research 2018 (2018)

19. Fearnhead, P., Taylor, B.M.: On estimating the ability of nba players. Journal of Quantitative analysis in sports 7(3) (2011)

20. Fomin, F.V., Lokshtanov, D., Saurabh, S., Zehavi, M.: Kernelization: Theory of Parameterized Preprocessing. Cambridge University Press (2019)

21. Gourvès, L., Lesca, J., Wilczynski, A.: Object Allocation via Swaps along a Social Network. In: 26th International Joint Conference on Artificial Intelligence (IJCAI’17), pp. 213–219. Melbourne, Australia (2017). https://doi.org/10.24963/ijcai.2017/31 https://hal.archives-ouvertes.fr/hal-0174519

22. Impagliazzo, R., Paturi, R.: On the complexity of $k$-SAT. J. Comput. Syst. Sci. 62(2), 367–375 (2001). https://doi.org/10.1006/jcss.2000.1727 http://dx.doi.org/10.1006/jcss.2000.1727

23. Impagliazzo, R., Paturi, R., Zane, F.: Which problems have strongly exponential complexity? J. Comput. Syst. Sci. 63(4), 512–530 (2001). https://doi.org/10.1006/jcss.2001.1774 http://dx.doi.org/10.1006/jcss.2001.1774

24. Irving, R.W.: Stable marriage and indifference. Discrete Applied Mathematics 48(3), 261–272 (1994)

25. Kinicki, A.J., Lockwood, C.A.: The interview process: An examination of factors recruiters use in evaluating job applicants. Journal of Vocational Behavior 26(2), 117–125 (1985). https://doi.org/https://doi.org/10.1016/0001-8791(85)90012-0 https://www.sciencedirect.com/science/article/pii/0001879185900120

26. Kolasa, T., Krol, D.: A survey of algorithms for paper-reviewer assignment problem. IETE Technical Review 28(2), 123–134 (2011). https://doi.org/10.4103/0256-4602.78092 https://www.tandfonline.com/doi/abs/10.4103/0256-4602.78092

27. Lian, J.W., Mattei, N., Noble, R., Walsh, T.: The conference paper assignment problem: Using order weighted averages to assign indivisible goods (2018), https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17396

28. Lokshtanov, D., Marx, D., Saurabh, S.: Slightly superexponential parameterized problems. SIAM Journal on Computing 47(3), 675–702 (2018). https://doi.org/10.1137/16M104834 https://doi.org/10.1137/16M104834

29. Mulder, J., de Bruijne, M.: Willingness of online respondents to participate in alternative modes of data collection. Surv. Pract 12(1), 1–11 (2019)

30. Nass, S.J., Levit, L.A., Gostin, L.O., et al.: The value and importance of health information privacy. In: Beyond the HIPAA Privacy Rule: Enhancing Privacy, Improving Health Through Research. National Academies Press (US) (2009)

31. Saban, D., Sethuraman, J.: The complexity of computing the random priority allocation matrix. Mathematics of Operations Research 40(4), 1005–1014 (2015). https://EconPapers.repec.org/RePEc:inm:ormoor:v:40:y:2015:i:4:p:1005-1014

32. Salch, S.M., Sugianto, S., Hilal, A., Ariansyah, D.: A study on the traffic impact of the road corridors due to flyover construction at surabaya intersection, banda aceh
of Indonesia. In: AIP Conference Proceedings. vol. 1903, p. 060005. AIP Publishing LLC (2017)

33. Singh, S., Dubey, G., Shrivastava, R.: A comparative analysis of assignment problem. IOSR Journal of Engineering 2(8), 01–15 (2012)

34. The Council for Higher Education and The Planning and Budgeting Committee: The higher education system in Israel (2014)

35. Topcu, M., Gulal, O.S.: The impact of covid-19 on emerging stock markets. Finance Research Letters 36, 101691 (2020).
https://doi.org/https://doi.org/10.1016/j.frl.2020.101691, https://www.sciencedirect.com/science/article/pii/S1544612320306966

36. Wu, K., DeVriese, A.: How students pick their housing situations: Factors and analysis (2016)

37. Zeren, F., HIZARCI, A.: The impact of covid-19 coronavirus on stock markets: evidence from selected countries. Muhasebe ve Finans İncelemeleri Dergisi 3(1), 78–84 (2020)

38. Zhou, L.: On a conjecture by gale about one-sided matching problems. Journal of Economic Theory 52(1), 123–135 (1990), https://EconPapers.repec.org/RePEc:eee:jetheo:v:52:y:1990:i:1:p:123-135

39. Oncan, T.: A survey of the generalized assignment problem and its applications. INFOR: Information Systems and Operational Research 45(3), 123–141 (2007). https://doi.org/10.3138/infor.45.3.123, https://doi.org/10.3138/infor.45.3.123