Article

Fair Assignment for Reserved Nucleic Acid Testing

Na Xie [1], Zhidong Liu [1], Xiqun (Michael) Chen [2,3,*] and Shen Li [4,*]

[1] School of Management Science and Engineering, Central University of Finance and Economics, Beijing 100081, China
[2] College of Civil Engineering and Architecture, Zhejiang University, Hangzhou 310058, China
[3] Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies, Hangzhou 310027, China
[4] Department of Civil Engineering, Tsinghua University, Beijing 100084, China
[*] Correspondence: chenxiqun@zju.edu.cn (X.C.); sli299@tsinghua.edu.cn (S.L.)

Abstract: Corona Virus Disease 2019 (COVID-19) is now treating the health of millions of people worldwide. The Chinese government now applies nucleic acid testing as a tool to detect patients from healthy people to control the spread of COVID-19. However, people may come to the nucleic acid testing stations simultaneously, leading to long queues and wasting time. In this paper, we proposed the reserved nucleic acid testing method, which could be easily implemented via Web applications associated with nucleic acid testing. Its key idea is to assign people to different pre-scheduled time slots so that the number of people arriving at a certain time slot can be controlled under the capacity, and thus congestion can be relieved. The key question is how to assign people in a fair manner. We propose a concise model to formalize and analyze the minimum total envy and pairwise fairness assignment problem for a variety of reservation-based applications, including nuclear acid testing. Its objective is to maximize the sum of each person's utility under the capacity constraints of time slots. The decision variables are the time slot assignment of each person. We show that the envy-freeness solution is usually unavailable. However, we can minimize the total envy through appropriate arrangements and realize pairwise fairness with equal-chance shuffling.

1. Introduction

Corona Virus Disease 2019 (COVID-19) can cause a series of illnesses, such as the common cold and severe acute respiratory syndrome. To deal with COVID-19, different measures had been taken [1]. One of these measures is influenza vaccination which can not only reduce the number of infected people, but also effectively reduce the mortality of severe patients with COVID-19. However, due to the high financial cost and the growing anti-vaccine movement, the vaccination rates in many countries remain low.

The Chinese government is now using another measure, nucleic acid testing, to detect and isolate patients from healthy people to control the spread of COVID-19 [2–4]. Nucleic acid testing is now worldwide accepted as the "gold standard" for the diagnosis of COVID-19 due to its high sensitivity and specificity. In April 2022, the Chinese government announced that the Chinese people had done 11.5 billion times nucleic acid testing (an average of 8 times per person).

After a large population of people is tested, the detected patients will be quarantined before they are cured [5]. The spread of the disease can be noticeably slowed down. For example, the Chinese government to set up thousands of nucleic acid testing stations in provincial capitals and cities to deal with the cluster outbreaks of the Omicron variant in many Chinese cities in March 2022. More than 10 million inhabitants take nucleic acid testing per day until outbreaks are finally controlled [6].

The success of such a measure lies in the exhaustive nucleic acid test for every needed people. However, the vast number of people in Chinese cities makes the exhaustive nucleic

acid test hard to carry out in practice. Although the Chinese government has established thousands of testing stations, people still need to wait in queues for a long time during rush hours. Figure 1 shows a long waiting queue outside a temporary nucleic acid testing station. This testing station is located in Tsinghua University, Beijing, China. Since it is impossible to build too many nucleic acid testing stations, people highly expect a better arrangement of their testing time schedule so as to avoid congestions.

Figure 1. A photo of a long waiting queue outside a temporary nucleic acid testing station.

This paper proposed a new model, aiming to determine all people's detailed testing plans through reservations before their departure. The orders are reasonably arranged during the peak demand period to avoid disorder and congestion by making reservations in advance. We propose a mathematical model to guarantee a fair assignment considering people's intentions. Specially, our objective is to maximize the sum of each person's utility under the capacity constraints of time slots. The decision variables are the time slot assignment of each person. We start from the simple cases in which each person is assumed to take the test individually at a single testing station. Then, we extended the model to consider group testing cases and multiple testing stations.

The fairness concept considered in this work is the envy-free assignments [7], where every person does not want to exchange its assigned time slot with any other agent's assigned time slot. However, we construct counterexamples to show that envy-freeness solution is usually unavailable. Furthermore, we can minimize total envy through appropriate arrangements and realize pairwise fairness with equal-chance shuffling. This indicates that the proposed model is useful in practice.

The rest of this paper is arranged as follows. In Section 2, we introduce the mathematical model of fair assignment for reserved transportation. In Section 3, we extend the model

to consider group testing cases and multiple testing stations. The definition of fairness is also discussed. In Section 4, we provide several numerical experiments to illustrate the new model. In Section 5, we conclude the whole paper and discuss future study directions.

2. The Fair Assignment Problem

2.1. The Basic Problem and the Solving Method

In this paper, the possible arrival times of people to a certain nucleic acid testing station are discretized into several time slots and are indexed by j, $j = 1, \ldots, m$. For each time slot, we can assume special capacity c_j (positive integers) to constrain the maximum number of people arriving at time slot j. We do not require all c_j to be equal.

There are n people to be scheduled. Each person has his/her own preference of time slots. Let us use a weighting vector $\mathbf{w}_i \in R^n$ to specify the time slot preference of people i, $i = 1, \ldots, n$. Such a weighting vector can be viewed as a person's utility for an assignment. If an element $w_{i,j}$ of vector $\mathbf{w}_i$ is larger than another element $w_{i,k}$, then the i th person prefers to arrive at time slot j rather than time slot k. We require $w_{i,j} \geq 0$ for $i = 1, \ldots, n$, $j = 1, \ldots, m$ and $\sum_{j=1}^{m} w_{i,j} = 1$, $\forall i$, to maintain a normalized weights of a person for different time slots.

As illustrated in Figure 2, the time slots are listed in chronological order. The preferences of two persons are illustrated with red color and blue color, respectively in bar plot style. In other words, each bar stands for the preference value $w_{i,j}$ that the i th person arrives at time slot j. The higher a bar is, the higher preference this person give its preference to a special time slot.
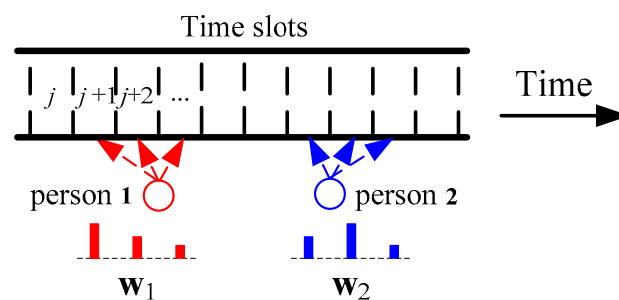

Figure 2. An illustration of the fair assignment problem.

We set a vector decision variable $\mathbf{x}_i \in R^n$ to indicate the assigned time slot for the i th person. Each element of vector $\mathbf{x}_i$ can be either 0 or 1. Since each person is unit-demand, it must be assigned to only one time slot. So, we have $\sum_{j=1}^{m} x_{i,j} = 1$, $\forall i$.

The fair assignment aims to maximize the total preferences/utilities of all people. The normality of the weighting coefficient vectors enables us to treat the preferences of different people equally and guarantee that their preferences/utilities are additive. So, we can formulate the basic mathematical programming model as follows:

$$\text{Problem (I)}: \ \max \sum_{i=1}^{n} \mathbf{w}_i^{\mathsf{T}} \mathbf{x}_i \tag{1}$$

$$\text{s.t.} \ \sum_{j=1}^{m} x_{i,j} = 1, \ \forall i \tag{2}$$

$$\sum_{i=1}^{n} x_{i,j} \leq c_j, \ \forall j \tag{3}$$

$$x_{i,j} \in \{0, 1\}, \ \forall i, j \tag{4}$$

We can also view the above problem as a special kind of assignment problem [8,9], where n people will be assigned to $\sum_{j=1}^{m} c_j$ time slots. We can use Hungarian algorithm [10] to solve Problem (I). However, when we consider some more complex scenarios (e.g., group reservation that will be discussed in Section 3.1), the assignment problem formulation will be quite complicated to analyze the envy property. So, in the rest of this paper, we will carry out our discussion in the formulation of Problem (I).

Noticing the intrinsic sparse structure of the decision variables (all elements of $\mathbf{x}_i$ should be 0 except one element), we can relax the binary variable constraints and obtain a relaxed mathematical programming model as follows:

$$\text{Problem (II)}: \ \max \sum_{i=1}^{n} \mathbf{w}_i^{\mathsf{T}} \mathbf{x}_i \tag{5}$$

$$\text{s.t.} \sum_{j=1}^{m} x_{i,j} = 1, \ \forall i \tag{6}$$

$$\sum_{i=1}^{n} x_{i,j} \leq c_j, \ \forall j \tag{7}$$

$$0 \leq x_{i,j} \leq 1, \ \forall i, j \tag{8}$$

Problem (II) is a linear programming problem and can be quickly solved by either the simplex method or the interior point method [11]. There are lots of mature software/toolbox that we can be used to solve such linear programming problems.

Moreover, the constraints coefficient matrix in Problem (II) is a unimodular matrix, and the right-hand side of the constraints are all integers. Similar to the proof for the totally unimodular matrix for bipartite graphs and assignment problem [12,13], we can prove that the constraints coefficient matrix for constraints (5)–(8) is totally unimodular. A linear programming problem (in standard form) with a totally unimodular coefficient matrix yields an optimal solution in integers for any objective vector and any integer vector on the right-hand side of the constraints [14,15].

So, the optimal solution of Problem (II) should also be one of the optimal solutions of Problem (I), and their optimal values are equal. It should be pointed out that Problem (I) may exist some different optimal solutions though their optimal values are equal.

Moreover, we can naturally derive the following lemmas.

Lemma 1. *Problem (I) has a feasible solution if and only if*

$$n \leq \sum_{j=1}^{m} c_j \tag{9}$$

Proof. The necessary condition is obvious. The sufficiency condition can be proved by constructing a feasible assignment solution as follows: assign the first c_1 people to time slot 1, then assign the first c_2 people of the remaining people to time slot 2, repeat such operations until all people have been assigned to a certain time slot. We can reach a feasible solution. □

Lemma 2. *For the optimal assignment obtained by solving Problem (I), there exits not a person (i.e., people i, $i = 1, \ldots, n$) who can be reassigned from time slot j to time slot k with $w_{i,j} < w_{i,k}$. Meanwhile, keep the capacity constraint of time slot k unviolated.*

Proof. We can easily prove this lemma by contradiction. If such a reassignment is valid and the capacity constraint of time slot k will not be unviolated, the new assignment is feasible and will also have a larger objective value, which contradicts our assumption of the optimal assignment. □

2.2. Envy-Freeness and Minimum Total Envy

The programming model formulated above maximizes the total preferences/utilities of all people. In the rest, we will analyze whether this leads to fair assignments of different people.

One frequently used definition of fairness is envy-freeness [7]. Focusing on this work, we define envy-freeness to mean: when the optimal assignment solution of Problem (I) is obtained, every person does not want to exchange time slots with any other person. More specially, we have the following definition.

Definition 1. *For an assignment, the ith person who is assigned to time slot j is said to envy another person assigned to time slot k, if $w_{i,j} < w_{i,k}$, $i = 1, \ldots, n$, $j, k = 1, \ldots, m$.*

Theorem 1. *Envy-freeness may not exist for the above problem setting.*

Proof. We can easily find a counterexample. Suppose there are two time slots available and two people (Person A and Person B) are competing for them. The maximum capacity of each time slot is just 1. If the weighting coefficient vector for Person A is $[0.8, \ 0.2]^T$, and the weighting coefficient vector Person B is $[0.6, \ 0.4]^T$; the optimal solution of Problem (I) should be assigning Person A to the first time slot and assigning Person B to the second time slot. However, Person B obviously envies Person A and wants to exchange with Person A. Even if the weighting coefficient vectors for two people are the same (say both $[0.8, \ 0.2]^T$), there exists no envy-free assignment solution, since only one person will be assigned to the first time slot. □

So, the fairness concept we focus on in this work is the minimum total envy assignment. It is hard to enumerate all possible exchange pairs for a certain problem setting and counter the values of total envy. So, we turn to examine the possible reduction of the total envy caused by transferring one assignment solution to another assignment solution. We have the following definition, lemma, and theorem.

Definition 2. *Group exchange operation. A feasible assignment solution can be transferred to another feasible assignment solution by exchanging a few people's time slots in a cyclic manner.*

For example, given three people (Person A, Person B, and Person C) assigned to three different time slots, we can get two possible exchanges: (a) Person A reassigned to Person B's previous time slot, Person B reassigned to Person C's previous time slot, and Person C reassigned to Person A's previous time slot; (b) Person A reassigned to Person C's previous time slot, Person B reassigned to Person A's previous time slot, and Person C reassigned to Person B's previous time slot.

Lemma 3. *Given any two feasible assignment solutions, we can always transfer from one assignment solution to another by first exchanging a few groups of people's time slots sequentially and then reassigning a few people's time slots to some under-crowed time slots.*

Proof. Comparing any two feasible assignment solutions, we can search whether there exists any possible exchange based on Lemma 1 and Lemma 2. As shown in Figure 3, we can view two assignment solutions as two groups of nodes, and each node denotes one person who has been reassigned to another time slot. Linking the nodes that represent the same person in two groups, we can get a number of edges that bridge two groups. Our search is to find the possible cycles that lie in these bridges. □

For example, we find an exchange group of four people in Figure 3. In Assignment Solution A, person 1 is assigned to the (j + 1)th time slot, person 2 is assigned to the (j + 2)th time slot, person 3 is assigned to the (j + 5)th time slot, person 4 is assigned to the (j + 6)th time slot. In Assignment Solution B, person 1 is assigned to the (j + 2)th time slot,

person 2 is assigned to the (j + 5)th time slot, person 3 is assigned to the (j + 6)th time slot, person 4 is assigned to the (j + 1)th time slot. So, we can switch the assignment of these four persons in Assugnment Solution A to formulate a different yet still feasible solution Assignment Solution B. Such a cycle search could be done by using the classical Hungarian algorithm (Kuhn, 1955) for the standard assignment problem. Since the search algorithm is beyond our main theme, we do not discuss it further.
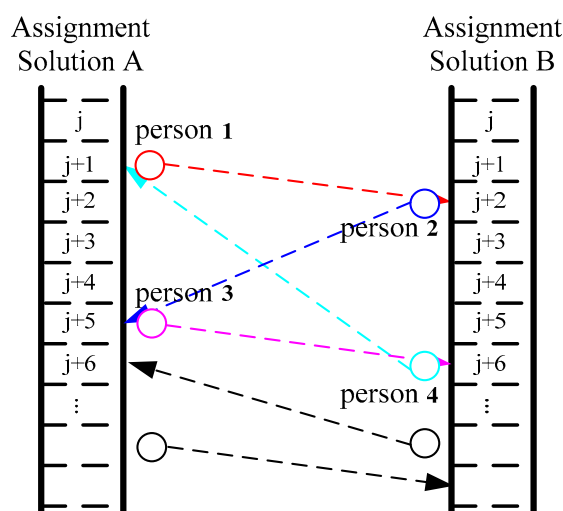

Figure 3. An illustration of the relationship between two feasible assignment solutions.

If we find such an exchange, we make all the encountered people an exchange group. Then, we repeat searching in the unmarked people until all possible exchange groups have been marked. Since the number of people is limited, the first-stage marking process ends in finite rounds. Then, we search whether there is a person in the yet unmarked people who have been reassigned individually. If such a person could be found, we make them reassigned. Then, we repeat searching in the unmarked people until all possibly reassigned people have been marked. Noticing that the number of people is limited, the second-stage marking process ends in finite rounds, too.

Definition 3. *Reduction of total envy. A feasible group exchange is said to reduce the total envy, if the number of encountered people who exchange with their envied people is larger than the number of the encountered people who become envious.*

Lemma 4. *For the optimal assignment obtained by solving Problem (I), no two people can exchange their time slots to simultaneously keep all people's total preferences/utilities unchanged and reduce their total envies.*

Proof. We still prove this lemma by contradiction. Suppose there are two people, say person p and person q, $p, q = 1, \ldots, n$, $p \neq q$. In the optimal assignment, they are assigned to time slot j and time slot k, $j \neq k$, respectively. So, we have

$$w_{p,j} + w_{q,k} \geq w_{p,k} + w_{q,j} \tag{10}$$

Otherwise, we can exchange them for a new assignment with a larger objective value.

On the other side, if these two people both envy each other's time slot and can be exchanged to reduce the total envy, we must have

$$w_{p,j} < w_{p,k} \text{ and } w_{q,k} < w_{q,j} \tag{11}$$

However, Inequality (11) contradicts Inequality (10), because of the cyclic manner. □

Lemma 5. *For the optimal assignment obtained by solving Problem (I), there is no feasible group exchange that encounters s people (s = 2, . . . , n) to simultaneously keep the total preferences/utilities of all people unchanged and reduce their total envies.*

Proof. The proof is quite similar to that for Lemma 2. Noticing the cyclic manner of a feasible exchange, we easily reach the conclusion. □

Theorem 2. *The optimal assignment obtained by solving Problem (I) leads to the minimum total envy. In other words, there are no other assignments to which can be transferred from the optimal assignment, and any of the related group exchanges can reduce the total envy.*

Proof. Suppose a few group exchanges exist between the optimal assignment and other different assignments. Based on Lemma 3, Lemma 4 and Lemma 5, neither of such group exchanges can reduce the total envy. □

Based on Theorem 2, we can see that the solution of Problem (II) guarantees to keep an appropriate envy balance between people.

2.3. Improve Pairwise Fairness between People

Lemma 2 indicates that we can keep relative fairness between two people in the optimal assignment solution, if Inequality (9) holds strictly unequal. However, let us consider two people, say people p and people q, $p, q = 1, \ldots, n$, $p \neq q$. In the optimal assignment, they are assigned to time slot j and time slot k, $j \neq k$, respectively. If we have $w_{p,j} = w_{q,j} > w_{q,k} = w_{p,k}$, we call these two people are identical-but-unequally-treated in this situation.

Clearly, exchanging these two people may change neither total preferences/utilities nor total envy. This can also be explained that multiple solutions may exist, yielding the same optimal value for Problem (I). However, assigning which people to time slot j may arise strong feelings of unfairness. So, we suggest introducing a random choice mechanism to determine which people will win a better time slot if we detect identical-but-unequally-treated people pair(s) in the optimal assignment solution in Problem (I).

Definition 4. *Pairwise fairness. An optimal assignment solution is said to have pairwise fairness if the assignment of any identical-but-unequally-treated people group to better time slots is randomly determined with an equal probability among people.*

To reach pairwise fairness, we provide the following algorithm (Algorithm 1), which is somewhat similar to the Fisher-Yates shuffle algorithm [16].

Algorithm 1: Shuffle Algorithm for Pairwise Fairness

Step 0: Suppose we have obtained the optimal assignment solution, and all people are sorted according to their indices. Set $p := 1$.
Step 1: Start from person p. Sequentially scan each of the $(p + 1)$ th to the n th remaining people to check whether we have an identical-but-unequally-treated people pair. Once we detect an identical-but-unequally-treated people pair, say person q, it generates a random number uniformly distributed between 0 and 1. If this random number is larger than 50%, we exchange the time slots assigned to person p and person q; otherwise, keep the current assignment. Go on until the rest of the people have been scanned.
Step 2: Set $p := p + 1$. If $p = n$, exit the algorithm; otherwise, go to Step 1 for another round of scan.

Assuming that we can generate a random number in $O(1)$ time, the above algorithm has a time complexity of $O(n^2)$.

Theorem 3. *The shuffle algorithm will not change the total envy, but the yielded optimal assignment solution has pairwise fairness.*

The proof is quite straightforward and is thus omitted here.

3. Extensions and Discussions

3.1. Group Reservation Extensions

The formulation of Problem (I) neglect the group reservations which might be common in practice. Indeed, some people have to take care of their dependent members of the family (older parents, disabled, kids, etc.). So, they will make a group reservation for nucleic acid testing.

To handle such cases, we set a vector decision variable $\mathbf{x}_i$ to indicate the assigned time slot for the i th group of people. So, we modify the mathematical programming model as:

$$\text{Problem (III)}: \ \max \sum_{i=1}^{n} g_i \mathbf{w}_i^{\mathrm{T}} \mathbf{x}_i \tag{12}$$

$$\text{s.t.} \sum_{j=1}^{m} x_{i,j} = 1, \ \forall i \tag{13}$$

$$\sum_{i=1}^{n} g_i x_{i,j} \leq c_j, \ \forall j \tag{14}$$

$$x_{i,j} \in \{0,1\}, \ \forall i,j \tag{15}$$

where g_i denotes the group size of the i th group of people and is a known constant for this mathematical programming model. When a person goes alone to take the nucleic acid testing, the group size is 1.

The coefficients in Inequality (14) does not formulate a totally unimodular coefficient matrix, if there exists any $g_i \geq 2$. Generally, we cannot reach a relaxed linear programming model that has the same optimal solution. So, we have to use the software/toolbox for integer programming to get the optimal solution.

It is easy to verify that Theorem 1 and Theorem 2 presented in Section 2.2 still hold for Problem (III). In other words, the solution of Problem (III) guarantees to keep an appropriate envy balance between people. In addition, we can still apply the Shuffle Algorithm for Pairwise Fairness proposed in Section 2.3, where only the identical-but-unequally-treated groups of people pair may be exchanged their time slots with equal probability.

3.2. Multiple Nucleic Acid Testing Stations Extensions

In the above formulated problems, only one testing station is considered, since most people will go to the nearest testing point. For example, during cluster outbreaks of the Omicron variant in many Chinese cities in March 2022, more than 10 million inhabitants go to the nearest testing station located within a 15-min walk from his/her home or office [6].

Sometimes, people may choose between several testing stations to find their faviorest time slot. To handle such cases, we set a vector decision variable $\mathbf{x}_i^{(k)} \in R^n$ to indicate the assigned time slot for the i th group of people to the k th testing station, $k = 1, \ldots, s$. For each time slot at the k th testing station, we can assume special capacity $c_j^{(k)}$ (positive integers) to constrain the maximum number of people arriving at time slot j.

The mathematical programming model considering both group reservation and station choice can be formulated as:

$$\text{Problem (IV)}: \ \max_{x_{i,j}^{(k)} \in \{0,1\}} \sum_{k=1}^{s} \sum_{i=1}^{n} g_i \mathbf{w}_i^{\mathrm{T}} \mathbf{x}_i^{(k)} \tag{16}$$

$$\text{s.t.} \sum_{k=1}^{s} \sum_{j=1}^{m} x_{i,j}^{(k)} = 1, \; \forall i \tag{17}$$

$$\sum_{i=1}^{n} g_i x_{i,j}^{(k)} \le c_j^{(k)}, \; \forall j, k \tag{18}$$

$$x_{i,j}^{(k)}, \; \forall i, j, k \tag{19}$$

where Equation (17) indicating each group of people must be assigned to only one time slot at one station; Inequality (18) indicating the number of people arrived at time slot j for he k th testing station should not exceed the maximum capacity.

Similarly, Theorem 1 and Theorem 2 presented in Section 2.2 still hold for Problem (IV). However, Problem (IV) cannot be relaxed into a linear programming model with the same optimal value. When the number of groups of people and the number of stations increase, the solving complexity of Problem (IV) increases significantly [9,17].

3.3. Discussions of Fairness

In this paper, fairness refers to distributing the benefits and costs to people and the degree to which the distribution is considered fair and appropriate. In some studies, researchers preferred to use 'equity' or 'justice' as alternatives to 'fairness', but their meanings are close [18–20]. We adopt the term 'fairness' here because our model of fairness is based on the concept of envy-free fairness studied for decades in the literature [7,12,21].

The difficulty of considering fairness in practice partly lies in the difficulty of defining appropriate fairness in both theory and applications. First, the definition and measure of equity in transportation vary significantly with respect to the applied scenarios [22,23]. For example, in the equity of ramp control problems [24], we can use either the weighted sum of the average delay at various on-ramps or the Gini coefficient drawn from Lorenz curves [25]. It is hard to find a common agreement of the criterion for fairness/equity.

As revealed in Arrow's Impossibility Theorem [26,27], there might be no perfect solution that makes every people satisfied. So, we need to define appropriate fairness that most people could accept. Such definition should lead to mathematical programming problems that can be quickly solved even for large-scale cases since we need to apply them to handle thousands of people.
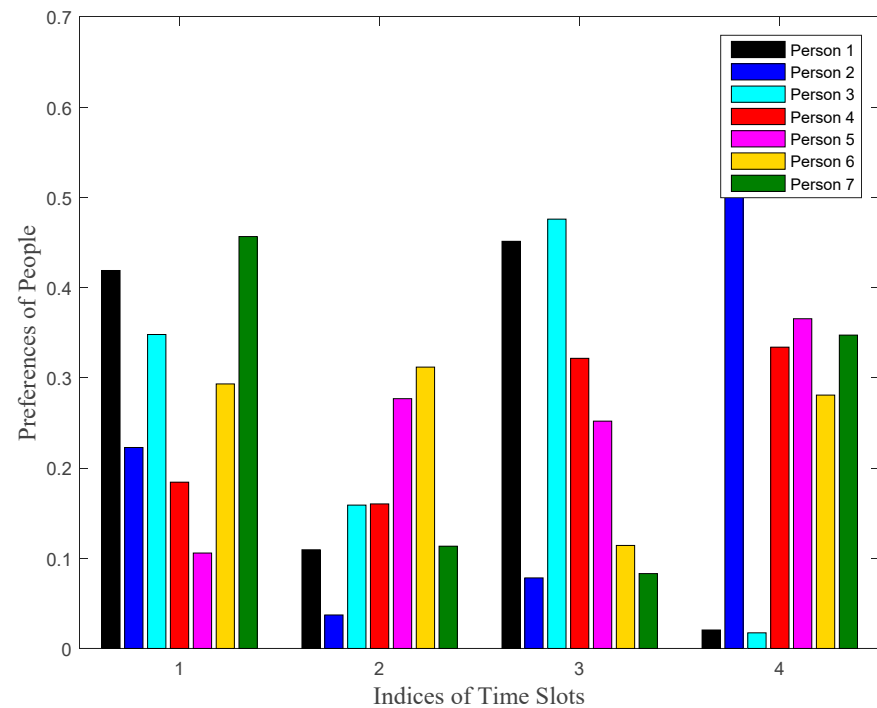
Fortunately, in this paper, we establish a definition of fairness that meets the above requirements. Moreover, the solution of our model simultaneously maximizes the utility and minimizes total envy. In other words, the maximum utility solution coincides with the fairest solution in our model. We expect more research interests will be drawn in this direction to fathom other definitions for fairness used in transportation.
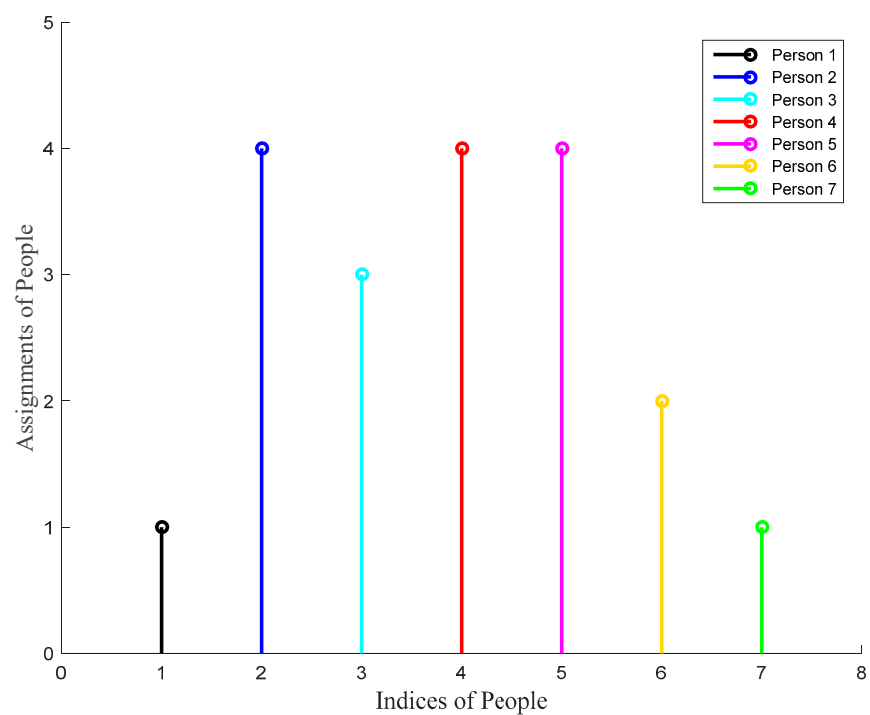
4. Testing Results

In this section, we present two illustrative numerical tests to demonstrate our algorithms. In the first example, we solve the basic Problem (I) via attacking its linear programming relaxation. In the second example, we consider the more complex Problem (IV).

In the first experiment, 7 people are waiting to be assigned to 4 time slots. The capacities of all time slots are set to 2, 2, 1, 3. The preference of each person is randomly generated and is plotted in Figure 4a after normalization. Figure 4b shows the obtained assignment solution after we solve the corresponding Problem (II). The preferences of people are illustrated in bar plot style. The higher a bar is, the higher preference this person give its preference to a special time slot. The assignments of people are illustrated in sterm plot styple. The little circles at tips of the vertical lines represent the exact value of the assignment for each person. We can see that Person 2, Person 3, Person 4, Person 5, Person 6, and Person 7 are assigned to their most preferred time slots, respectively. People 1 is assigned to its second preferred time slots; since its most preferred time slot has reached capacity. Moreover, the total envy of this solution is 1 (only People 1 envies People 3). It is easy to check that we cannot exchange People 1 with People 3 to reduce the total envy.

This result indicates that not every person can be assigned to its favorist time slot. As indicated in Theorem 1, even for a randomly generated scenario, envy-freeness may not exist. However, the above solving algorithm can achieve the minimum total envy.

(**a**)

(**b**)

Figure 4. A numerical test of the proposed model: (**a**) the preferences of people; (**b**) the assignments of people.

We do not provide large scale numerical testing results for the original problem, simply because it is unnecessary. Simplex algorithms and Interior Point Method can easily solve solved the original problem with thousands of people and less than 100 time slot choices. Moreover, we do not need any a nucleic acid testing station to handle over 10 k people in our daily life, since crowded nucleic acid testing stations are prone to aggregated infections.

In the second experiment, we assume there exist 3 groups of people. The group sizes are 2, 2, and 1, respectively. We further assume that there are two testing stations, each of which has four time slots available. The capacities of the times slots are 2, 2, 1, and 1, respectively, for both two testing stations. The preference weighting vector of Group 1 for the first station is (0.3, 0.4, 0.1, 0.2); the preference weighting vector of Group 1 for the second station is (0.4, 0.3, 0.2, 0.1). The preference weighting vector of Group 2 for the first station is (0.3, 0.2, 0.4, 0.1); the preference weighting vector of Group 2 for the second station is (0.2, 0.3, 0.1, 0.4). The preference weighting vector of Group 3 for the first station is (0.25, 0.25, 0.3, 0.2); the preference weighting vector of Group 3 for the second station is (0.25, 0.25, 0.25, 0.25). We formulate an integer programming problem and solve it by using Gurobi 9.0. The optimal solution indicates that Group 1 will be assigned to time slot 2 of station 1; Group 2 will be assigned to time slot 2 of station 2; and Group 3 will be assigned to time slot 3 of station 1. This indicates that our model can handle both group reservation and multiple station choice that may be pervasive in practice.

We solved the extended problem with tens of groups of people and less than 10 time slot choices, by using Gurobi 9.0 academic license within 10 s. However, when the number of groups of people, time slots, and available nucleic acid testing stations ever increases, the solving process becomes too slow.

5. Conclusions

To the best of our knowledge, the model proposed in this paper is the first mathematical model for reserved nucleic acid testing. We highlight reservation fairness. Our studies reveal that we can simultaneously achieve maximum utility and minimum total envy. These results set up a basis for future studies on complex fair assignment problems considering more issues, e.g., time-varying demand/supply, fluctuations, and randomness.

Such a reservation-based assignment plan can be easily carried out in practice. Since mobile phones are pervasive now, we can establish software on people's mobile phones and let every person characterize its preferences on its mobile phone conveniently. The software can collect all the information and calculate the optimal assignment plan on the cloud. With the aid of wireless communication, mobile application, and cloud computing, we can provide refined management for some complex social problems.

In addition, we do not further discuss the solving algorithm for the integer programming problem formulated in this paper. It is expected that these problems can be solved in a more efficient way by using column generation method [28–30]. Constrained by the length limit, we would like to discuss this issue in a dedicated paper.

It should be pointed out that our focus of this paper is to find the best plan for people under given nucleic acid testing supplies. Indeed, governments may use different measures to further control the waiting queue length around nucleic acid testing stations. For example, how to select the locations of testing stations remains to be further studied. The capacities of different time lots can also be optimized to better serve the testing demand under given financial budgets.

Author Contributions: Data curation, X.C.; Methodology, S.L.; Writing—original draft, Z.L.; Writing—review & editing, N.X. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

References

1. Benita, F. Human mobility behavior in COVID-19: A systematic literature review and bibliometric analysis. *Sustain. Cities Soc.* **2021**, *70*, 102916. [CrossRef] [PubMed]
2. Au, W.Y.; Cheung, P.P.H. Diagnostic performances of common nucleic acid tests for SARS-CoV-2 in hospitals and clinics: A systematic review and meta-analysis. *Lancet Microbe* **2021**, 2, e704–e714. [CrossRef]
3. Esbin, M.N.; Whitney, O.N.; Chong, S.; Maurer, A.; Darzacq, X.; Tjian, R. Overcoming the bottleneck to widespread testing: A rapid review of nucleic acid testing approaches for COVID-19 detection. *RNA* **2020**, *28*, 771–783. [CrossRef] [PubMed]
4. Zhang, S.; Su, X.; Wang, J.; Chen, M.; Li, C.; Li, T.; Ge, S.; Xia, N. Nucleic acid testing for coronavirus disease 2019: Demand, research progression, and perspective. *Crit. Rev. Anal. Chem.* **2022**, *52*, 413–424. [CrossRef] [PubMed]
5. Wang, Y.-J.; Xue, J.-H.; Fang, Z.-X.; Xie, J.-W.; Niu, J.-J.; Yang, T.-C.; Lin, L.-R. A 14+7 day quarantine period and a dual nucleic acid testing reagent strategy detect potentially indiscoverable Coronavirus disease 2019 infections in Xiamen, China. *Clin. Chim. Acta* **2022**, *532*, 89–94. [CrossRef] [PubMed]
6. Li, Z.; Gao, J. 15-Minute nucleic acid test circles strategy in large cities in China. *J. Biosaf. Biosecurity* **2022**, *4*, 84–85. [CrossRef]
7. Foley, D.K. Resource Allocation and the Public Sector. *Yale Econ. Essays* **1967**, *7*, 45–98.
8. Haddadi, S.; Gattal, E. Combining data reduction, MIP solver and iterated local search for generalized assignment. *Int. J. Manag. Sci. Eng. Manag.* **2022**, *17*, 93–102. [CrossRef]
9. Öncan, T. A survey of the generalized assignment problem and its applications. *INFOR Inf. Syst. Oper. Res.* **2007**, *45*, 123–141. [CrossRef]
10. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [CrossRef]
11. Luenberger, D.G.; Ye, Y. *Linear and Nonlinear Programming*, 4th ed.; Springer: Berlin/Heidelberg, Germany, 2016.
12. Aigner-Horev, E.; Segal-Halevi, E. Envy-free matchings in bipartite graphs and their applications to fair division. *Inf. Sci.* **2022**, *587*, 164–187. [CrossRef]
13. Asratian, A.S.; Denley, T.M.; Häggkvist, R. *Bipartite Graphs and Their Applications*; Cambridge University Press: Cambridge, UK, 1998.
14. Hoffman, A.J.; Kruskal, J.B. Integral boundary point of convex polyhedral. In *Linear Inequalities and Related Systems*; Chapter 13; Kuhn, H.W., Tucker, A.W., Eds.; Princeton University Press: Princeton, NJ, USA, 1956.
15. Veinott, A.F., Jr.; Dantzig, G.B. Integral extreme points. *SIAM Rev.* **1968**, *10*, 371–372. [CrossRef]
16. Fisher, R.A.; Yates, F. *Statistical Tables for Biological, Agricultural and Medical Research*, 3rd ed.; Oliver & Boyd: London, UK, 1938; pp. 26–27.
17. Schreiber, E.L.; Korf, R.E.; Moffitt, M.D. Optimal multi-way number partitioning. *J. ACM* **2018**, *65*, 1–61. [CrossRef]
18. Ahmed, Q.I.; Lu, H.; Ye, S. Urban transportation and equity: A case study of Beijing and Karachi. *Transp. Res. Part A Policy Pract.* **2008**, *42*, 125–139. [CrossRef]
19. Litman, T. Evaluating transportation equity. *World Transp. Policy Pract.* **2002**, *8*, 50–65.
20. Sanchez, T.W.; Stolz, R.; Ma, J.S. *Moving to Equity: Addressing Inequitable Effects of Transportation Policies on Minorities*; The Civil Rights Project at Harvard University: Cambridge, MA, USA, 2003.
21. Brams, S.J.; Taylor, A.D. *Fair Division: From Cake-Cutting to Dispute Resolution*; Cambridge University Press: Cambridge, UK, 1996.
22. Liang, X.J.; Guler, S.I.; Gayah, V.V. An equitable traffic signal control scheme at isolated signalized intersections using Connected Vehicle technology. *Transp. Res. Part C Emerg. Technol.* **2019**, *110*, 81–97. [CrossRef]
23. Tian, Q.; Huang, H.-J.; Yang, H.; Gao, Z. Efficiency and equity of ramp control and capacity allocation mechanisms in a freeway corridor. *Transp. Res. Part C Emerg. Technol.* **2012**, *20*, 126–143. [CrossRef]
24. Kotsialos, A.; Papageorgiou, M. Efficiency and equity properties of freeway network-wide ramp metering with AMOC. *Transp. Res. Part C Emerg. Technol.* **2004**, *12*, 401–420. [CrossRef]
25. Yin, Y.; Liu, H.; Benour, H. A note on equity of ramp metering. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*; IEEE: Washington, DC, USA, 2004; pp. 497–502.
26. Arrow, K.J. A Difficulty in the Concept of Social Welfare. *J. Political Econ.* **1950**, *58*, 328–346. [CrossRef]
27. Arrow, K.J.; Sen, A.; Suzumura, K. (Eds.) *Handbook of Social Choice and Welfare*; Elsevier: Amsterdam, The Netherlands, 2002; Volume 1.
28. Wilhelm, W.E. A Technical Review of Column Generation in Integer Programming. *Optim. Eng.* **2001**, *2*, 159–200. [CrossRef]
29. Wolsey, L. *Integer Programming*, 2nd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2020.
30. Zeighami, V.; Soumis, F. Combining Benders' decomposition and column generationfor integrated crew pairing and personalized crew assignment problems. *Transp. Sci.* **2019**, *53*, 1213–1499. [CrossRef]