Existence of Pairwise Stability in Two-sided Matching Markets

by

Yasir Ali

A thesis submitted to the

Centre for Advanced Mathematics and Physics,
National University of Sciences and Technology,
H-12, Islamabad, Pakistan

for the degree of Doctor of Philosophy

Aug 2011

Existence of Pairwise Stability in Two-sided Matching Markets

by

Yasir Ali

A thesis
submitted for the degree of Doctor of Philosophy
in
Mathematics

Supervised by
Rashid Farooq

Centre for Advanced Mathematics and Physics,
National University of Sciences and Technology,
H-12, Islamabad, Pakistan

Abstract

In this thesis, we study two types of two-sided matching markets. The prime objective in these markets is to show that there always exists a pairwise stable matching. In these markets, each participant has a preference list. This preference list contains participants of the opposite side listed in an order. In these two-sided matching markets participants of one side can exchange money with participants of the opposite side. The preferences of the participants depend upon the money which they exchange. In fact, the preferences are given in terms of increasing functions of money.

First, we consider a one-to-many matching market. For this market we assume that the preferences are continuous, strictly increasing linear functions of money. We develop an algorithm to show the existence of pairwise stability in this matching model. This matching model is more general than that of marriage model by Gale and Shapley, assignment game by Shapley and Shubik and hybrid models by Eriksson and Karlander, Sotomayor and Farooq.

We also consider a market in which money appears as a discrete variable. For this market, we consider the preferences as strictly increasing functions of money. We use algorithmic approach to show the existence of pairwise stable outcome for the one-to-one matching market. This market generalizes the marriage model by Gale and Shapley.

Acknowledgement

I have no words to thank Allah Almighty for His uncountable blessing upon me and for giving me the strength and health to do this work.

I am very grateful to Dr. Rashid Farooq, my PhD supervisor, for introducing me this subject and taught me how to do research.

It is my pleasure to thank Prof. Asghar Qadir for introducing and motivating me to work in the field of optimization theory while I was doing my MSc at Quaid e Azam university, Islamabad, Pakistan. I wish to express my sincere gratitude to Prof. Muneer Ahmad Rasheed and specially Dr. Tayyab Kamran. Without their guidance, encouragement and support it would not be possible for me to complete this monograph. I would also like to thank my wife Dr. Quanita Kiran for her useful comments to improve this monograph. This research bears on imprint of many batch fellows, colleagues and staff who render their help during my PhD. Financial support of NUST is gratefully acknowledged.

Last but not least I wish to avail this opportunity, to express a sense of gratitude and love to my beloved parents, co-operative sisters, my cousin Khurram Shahazad and other relatives for their manual support, strength, help and for everything.

YASIR ALI

Preface

Matching problems arise in a variety of practical applications. For example, pattern matching is of great importance. In fact pattern matching is a computerized search operation which is used to find a given pattern within a database. Such a matching plays an ever increasing role in almost all aspects of our every day lives. Matching is commonly used in the fields of science and information processing and for different routine tasks including text editing, term rewriting, symbol manipulation, code generation, spelling correction, bibliographic search and natural language processing. Matching is also used in internet search engines like Google and Yahoo, and also in wireless local area networks and optical character recognition. Advanced methods and algorithms of matching today are used in genetics for locating DNA sequences, in criminology for fingerprint assessment, in geology for soil and in business for job allocations. Matching theory has been especially influential in economics, where it has been used to describe the formation of new jobs, as well as to describe other human relationships like marriage.

In this thesis we consider matchings in which set of participants are divided into two disjoint subsets; the set of individuals and the set of institutions. Generically, we recognize the individuals as workers and the institutions as firms. These types of matchings are called two-sided matching markets. Main question in two-sided matching markets is that of stability which entirely depends upon the preferences expressed by the participants. The concept of two-sided matching markets is well known in mathematical economics. In 1962, Gale and Shapley [23] introduced and solved the problem of stable two-sided matching. In their pioneering work, they presented the marriage model; a model in which a participant on one side is matched with at most one participant on the opposite side. The monetary transfer is not permitted in their model. For this reason, the participants in this model are called "rigid". Gale and Shapley proposed an algorithm which finds a stable marriage and also extended their work to college admissions problem (one-to-many matching model). The one-to-one buyer-seller model by Shapley and Shubik [46], known as assignment game, in contrast to Gale and Shapley's marriage model [23], deals with the participants who can trade money, that is, the "flexible" participants. Shapley and Shubik showed that the core of the assignment game is a non-empty complete lattice, where the core of a game is defined as the set of un-dominated outcomes. These problems have proven to be of unending interest to researchers from a wide range of communities, including mathematics, economics, computers sciences, management sciences and game theory.

Knuth [29], summarized most of what was known about the two-sided matching problem up until that time. He also mentioned some open problems concerning both algorithmic analysis issues and structural issues in stable marriage.

Crawford and Knoer [7] developed an algorithm, called the "salary adjustment process" which is a generalization of Gale and Shapley's deferred-acceptance algorithm to the case where money is present. The non-emptiness of the core is shown in this model [7]. A generalization of the assignment game [46] is also presented in Demange and Gale [11] where preferences of the participants may be represented by any continuous utility functions in the money variable.

Kaneko [26] gave a very general and complicated model. He unified the Gale and Shapley marriage model [23] and the Shapley and Shubik assignment game

[46] and established the non-emptiness of the core but did not establish the lattice property. The unification of the marriage model [23] and the continuous model of Demange and Gale [11] can be observed in Roth and Sotomayor [42]. However, the existence of stable outcome is not guaranteed in their model but they investigated the lattice property for payoffs in the core. A one-to-one matching model is proposed by Eriksson and Karlander [14] where they unified the discrete (marriage model [23]) and continuous (assignment game [46]) models. The marriage model [23] becomes a special case of their model if the participants, at least on one side, are rigid. The assignment game [46] is obtained when all participants are flexible. The existence of stable matching is guaranteed in this model [14]. They further discussed the lattice property of the set of stable outcomes. Some more investigation of their model is also found in Sotomayor [48]. Recently, Sotomayor [49] presented a one-to-one matching model, which is a special case of the model of Eriksson and Karlander [14] in the sense that all participants on one side are flexible. On the other side, however, some participants are rigid and the remaining are flexible. A characterization of the core of this hybrid market can be seen in this paper [49].

Motivated by the works of Eriksson and Karlander [14] and Sotomayor [48], Fujishige and Tamura [18] proposed a common generalization of the marriage model and the assignment game by utilizing the framework of discrete convex analysis developed by Murota [30–32]. They further extended their model in [19] by assuming possibly bounded side payments and proved the existence of pairwise stable outcome. The structure of the set of pairwise stable outcome is not discussed in their paper [19]. Farooq [15] gave a generalization of the models of Eriksson and Karlander [14] and Sotomayor [48] by identifying the preferences of the participants by strictly increasing linear valuations.

In mathematical economics markets with indivisible goods have been widely studied. Kelso and Crawford [27] presented a two-sided matching model with money and introduced a Gross Substitutes (GS) condition. Under GS they showed the non-emptiness of the core. Their model also includes the marriage model by Gale and Shapley [23] and the assignment game by Shapley and Shubik [46]. Gul and Stacchetti [25] further investigated the model of Kelso and Crawford and proposed two conditions: Single Improvement (SI) condition and No Complementarities (NC) condition. They showed that SI and NC are equivalent to GS for set functions. Fujishige and Yang [20] gave a relationship between GS and $M^\natural$-convexity for set functions. Quinzii [35] considered a model of an exchange economy with two kinds of goods. The first kind of good is perfectly divisible (money) and the other is indivisible. Each participant has a certain amount of money and at most one indivisible good. She proved that the core of the economy is non-empty. Gale [24] considered the model of Quinzii [35] and gave a direct proof of the existence of equilibrium by using a generalization of the lemma of Knaster, Kuratowski and Mazurkewicz [28] in combinatorial topology.

The literature on exchange of indivisible goods finds natural application in the exchange of kidneys for transplant. Roth, Sönmez and Ünver [40], addressed the issue of kidney transplant when there are constrains on the size of exchanges. The theoretical groundwork for kidney exchange without a constraint on the size of exchanges, explored in Roth Sönmez and Ünver [39], was laid in the papers by Shapley and Scarf [45], Roth and Postlewaite [37], and Abdulkadiroğlu and Sönmez [1].

The first chapter is devoted for the introduction of basic stable matching models, that is, marriage model [23] and assignment game [46]. In this chapter we also present some notations and definitions from the graph theory. A brief history of the two-sided stable matching is included in this chapter.

In Chapter 2, we consider a one-to-many matching market. For this market we assume that the valuations are strictly increasing linear functions of money. We show that pairwise stability exists for such markets. We develop an algorithm to show the existence of stable outcome for this model. To understand the working of the algorithm we also present an example. This matching model is more general than that of marriage model [23] by Gale and Shapley, assignment game [46] by Shapley and Shubik and hybrid models [14, 15, 48].

In Chapter 3, unlike the model of Farooq [15], we use money as a discrete variable rather than as a continuous variable. Such conditions are empirical since very small increments in the price may not be worthwhile for the participants. We define the pairwise stability for this model. We give a constructive proof to show the existence of a stable outcome in our model. An example is also included to understand the model and working of algorithm. At the end we discuss the termination and correctness of algorithm.

In Chapter 4, we present the summery and discuss some open problems related the models discussed in Chapter 2 and 3.

Contents

Chapter 1

Preliminaries

1.1 Introduction

Stable matching theory has been widely used in describing two-sided economics, like formation of marriages among unmatched individuals, relationships between unemployed workers and vacancies opened by firms or dealing between buyers and sellers in auctions. For this reason, there is a vital role of the stable matching theory especially in the fields of mathematical economics and game theory. In addition to this, the stable matching algorithm is also an active area of research. Stable matching algorithms connect algorithmic theory and combinatorial optimization. An important tool to investigate these algorithms is the graph theory which also has an extensive use in combinatorial optimization. We give a brief description of the stable matching problem and basic definitions for two-sided matching problem in Section 1.2. In Section 1.3, we discuss some well known models on the stable matching and highlight some approaches used in these models. A major innovation in mathematical economics is the use of methods stemming from graph theory to study relations between economic participants. Section 1.4 is devoted for the study of basic graph theory which is useful in subsequent chapters of this thesis.

1.2 Stable matching problem

In this section, we define basic terminology and notation for the stable matching problem (SMtP). Also, we review some basic stable matching models and their important features. We start by defining stable matching problem.

Definition 1.1. Given two disjoint sets, the *matching problem* is to assign elements of one set to elements of other set such that some criteria of desirability is optimized.

In any game-theoretic analysis it is very important to keep clearly in mind the rules of the game, as these will influence every aspect of the analysis. In matching markets, elements of the disjoint sets are matched to each other under these rules. We suppose that the general rule for governing matching is:

"Any pair of participants, from the opposite sets, who both consent to match with one an-other may do so, and every one is free to withhold its consent and remain unmatched". We will describe the rules in more detail at various points, in our discussion, later.

Typically, a stable matching is subjected to constraints involving capacities and preference list. The capacity of a participant is the maximum number of participants of the opposite set that can be assigned to it. It is usually called the *quota* of the participant. The preference list of a participant contains participants of the opposite set listed in such a way that the favorite participant appears prior to the least favorite one. Individual's preferences play a very critical role in game-theoretic models particularly and in economic models generally.

Definition 1.2. An *individual's preferences* are meant to represent how an individual would choose among different participants, if the individual were faced with a choice.

When an individual have to choose among different participants then obviously the individual may like a participant more than the others and may equally like some participants.

Definition 1.3. An individual *prefers* a participant a to a participant b, means that, if an individual were faced with a choice between the two, the individual would choose a and not b, and if faced with a choice from a set of participants that includes b, then it would not choose b if a were also available.

Definition 1.4. An individual is *indifferent* between two participants, means that, the individual might choose either one. We will say an individual likes a at least as well as b if the individual either prefers a to b or is indifferent between them.

Definition 1.5. If an individual is not indifferent between any two participants then the preferences of such individual are called *strict preferences.*

There are two assumptions, which are very important in the making of an individual's preference list from the economical point of view.

- The first condition on the preferences is that; the preferences must form a *complete ordering.*

This means that any two participants can be compared, that is, an individual may be indifferent between two participants but the individual is never confronted with a choice the individual is unable to make.

- The preferences of each individual must be *transitive.*

This means that if a participant a is liked at least as well as a participant b, and b is liked at least as well as a participant c then a is liked at least as well as c.

To express these preferences concisely, we consider two disjoint sets M and W, which are given as:

$$M = \{m_1, m_2, \ldots, m_{k'}\}, \qquad W = \{w_1, w_2, \ldots w_k\},$$

where k' and k are any finite numbers such that $|M| = k'$ and $|W| = k$

The preferences of each $m \in M$ is represented by an ordered list $Pl(m)$ on the set $W \cup \{m\}$. A preference list of m might be of the form

$$Pl(m) = w_2, w_1, w_5, m, w_3, \ldots, w_k, \tag{1.1}$$

where m's first choice is w_2, second and third choices of m are w_1 and w_5, respectively. If neither w_2 nor w_1 nor w_5 are among participants then m will choose to remain single.

Definition 1.6. Participants that appear before m in the $Pl(m)$ are said to be *acceptable* to m and rest are not acceptable to m for matching.

As in (1.1), $\{w_1, w_2, w_5\}$ are acceptable to m, and $W \setminus \{w_1, w_2, w_5\}$ are not acceptable. Similarly, participants which appear before w in the $Pl(w)$ are said to be *acceptable* to w.

Definition 1.7. If both m and w are acceptable to each other then they are said to be *mutually acceptable.*

In general, it is not possible to match each participant of the game, therefore, some participants may remain single.

Definition 1.8. The participants that could not find any match are called *unmatched* or *self-matched.*

Let m be indifferent between w_1 and w_5 then $Pl(m)$ assumes the form

$$Pl(m) = w_2, [w_1, w_5], m, w_3, \ldots, w_k.$$

We now discuss pairwise stable matching. Among the set of all possible matchings we are interested in such a matching that satisfies two important conditions. First condition requires that every pair of matched participants is acceptable to one another. Second condition demands that there should be no incentives for any pair which favor both the participants to form a new matching, by breaking up the current matching. The pair for which there exist such inducements is called blocking pair.

Definition 1.9. A *blocking pair* is a pair of participants that are not matched to each other in a matching but both prefers one another to their current assignees.

Definition 1.10. A matching is said to be *stable* if participants of each matched pair are mutually acceptable and the matching has no blocking pair.

The simplest example of stable matching problem is stable marriage problem (SMrP) due to Gale and Shapley [23]. In the marriage model [23], the preferences are strict and each participant is assigned to exactly one participant of the opposite set.

1.2.1 Stable marriage problem

In 1962, Gale and Shapley [23] initiated two-sided matching problem. An instance of the SMrP consists of two disjoint sets of size k, that is, M and W given by

$$M = \{m_1, m_2, \ldots, m_k\}, \qquad W = \{w_1, w_2, \ldots, w_k\}.$$

Here M denotes the set of men and W denotes the set of women. It is assumed that the participants of the opposite sets are mutually acceptable and each participant has a strict preference list. Therefore, each man ranks k women in a strictly ordered list, and similarly each woman has a strictly ordered preference list of k men. An example of SMrP of instance $k = 4$ can be seen in Table 1.1.

Preference profile of all 8 participants involve in the game are shown. In any list, a participant that appears at the first position from the left is the most preferred among the others. A participant that appears at the end of list is the least preferred one. For example the positions of w_3 and w_4 in the preference list of m_2 indicate that w_3 is the most preferred and w_4 is least preferred by m_2. While the position of w_1 indicates that w_1 is less preferred than w_3 and more preferred than w_2 by m_2. Similarly, we can explain the preference lists of other participants. For

Men's list	*Women's list*
$Pl(m_1) : w_1, w_2, w_3, w_4$	$Pl(w_1) : m_1, m_4, m_3, m_2$
$Pl(m_2) : w_3, w_1, w_2, w_4$	$Pl(w_2) : m_2, m_3, m_4, m_1$
$Pl(m_3) : w_1, w_3, w_4, w_2$	$Pl(w_3) : m_3, m_2, m_1, m_4$
$Pl(m_4) : w_1, w_2, w_4, w_3$	$Pl(w_4) : m_1, m_4, m_2, m_3$

Table 1.1: An instance of size $k = 4$

such an instance, a matching is a one-to-one correspondence between the men and women. Due to resemblance of this matching model with marriage of man and woman in real life, this model is also called the *marriage model.*

Definition 1.11. A *marriage*, denoted by X, is a set of man-woman pairs such that each man and woman appear only once.

Definition 1.12. A man and a woman are called *partner* in X if they are matched in X.

Note here that since there are equal number of men and women and all men and women are acceptable to each other, therefore, none of the participant will remain unmarried. Also since all men and women are acceptable to each other, it follows that the first requirement for a stable marriage holds true. Now we shall focus on blocking pair to study the existence of stable marriage.

Definition 1.13. If (m, w) is a blocking pair for a marriage X then we say that the man m and the woman w *block* X.

Definition 1.14. A marriage which contains a blocking pair is said to be *unstable marriage.*

In general, it may not be obvious from the statement of the problem that a stable marriage always exist, or how to find a stable marriage from a given instance. For example, consider the marriage instance with preference profile given in Table 1.1 and the assignment X^* defined by

$$X^* = \{(m_1, w_2), (m_2, w_3), (m_3, w_4), (m_4, w_1)\}. \tag{1.2}$$

Since each participant appears only once, therefore, X^* is a marriage. We check whether X^* is stable or not. This can be answered by checking whether there exists any blocking pair for the marriage X^*. Here we discuss the criteria for the existence of a blocking pair for a given marriage with a given preference profile. For this procedure, it suffices to consider participants of one set, say the set of men.

Suppose that for a given preference profile, X denotes a marriage of men and women. Then for each man $m \in M$, we check only those women $w \in W$ to whom m likes more than his partner in X. If there exists such a woman say w' to whom m likes more than his partner in X and w' prefers m to her partner in X then the pair (m, w') is a blocking pair for the marriage X. This yields that the marriage X is unstable.

Now we apply this procedure to check the stability of X^* for the preference profile given in Table 1.1. It is easy to see that (m_1, w_1) forms a blocking pair for X^* and, hence, the marriage X^* is unstable.

We saw that every marriage for an instance may not be a stable marriage. Thus it is natural to ask a question that can there be any instance for which there does not exist any stable marriage?

1.2.2 Gale-Shapley (GS) algorithm

In their paper [23], Gale and Shapley developed a fundamental theorem that there always exist a stable marriage in an instance of the SMrP. To prove this result they proposed an algorithm called *Gale-Shapley (GS) algorithm*. GS algorithm uses a deferred acceptance strategy, comprising proposals and rejections. Before the start of algorithm each participant is said to be free or unmarried. At any stage during the execution of GS algorithm, each participant is either engaged or free; each proposer may alternate between being engaged and being free. There are two possible orientations of GS algorithm, namely the man-oriented and the woman-oriented, depending on who makes proposals. In the man-oriented algorithm, each man proposes the first woman on his list to whom he has not proposed previously. If the woman is free, then she becomes engaged to a man who proposes her. If a woman receives more than one proposals then she accepts the most favorite one and rejects the remaining proposals. Let a man m proposes to an engaged woman w. If w prefers m to her current partner m' then she rejects m'. In such a case, m' becomes free and w becomes engaged with m. Otherwise, w rejects m and remains engaged with m'. In this case m remains free. This process ends when every men get married. The process is similar for the woman-oriented algorithm, but the proposals are then made by the women.

Now we present man-oriented algorithm formally.

Algorithm `GS_Allocation`

Step 1: Initially, set all participants to be free. Each man proposes the first woman in his preference list. Each proposed woman becomes engaged with her most preferred man who proposes her and rejects other proposers. Rejected men become free.

Step 2: If there is no free man then Stop.

Step 3: Each free man proposes the first woman in his preference list to whom he has not proposed previously. Each proposed woman becomes engaged with her most preferred man who proposes her and rejects other proposers. Rejected men become free. Go to Step 2.

As expressed above, the `GS_Allocation` involves an element of non-determinism, since the order in which the free men (proposers) propose is not given. However, Gusfield and Irving [22] noted that, whatever is the order of the proposals, the resulting marriage is always the same.

The fundamental nature of the `GS_Allocation` is summarized in the following theorem.

Theorem 1.15 (Gale and Shapley [23])**.** *For any given instance of the SMrP, the* `GS_Allocation` *terminates and the engaged pairs constitute a stable marriage on termination.*

Definition 1.16. For any marriage market, a stable marriage X is called *man-optimal* if every man likes his partner in marriage X at least as well as in any other stable marriage. Similarly, a stable marriage X is called *woman-optimal* if every woman likes her partner in marriage X at least as well as in any other stable marriage.

In man-oriented algorithm, each man obtains his best possible stable partner and each woman obtains her worst possible stable partner. Equivalently, when the woman-oriented algorithm is applied, each woman gets her best possible stable partner while each man gets his worst possible stable partner. We can present this property of GS algorithm more precisely as:

Theorem 1.17 (Gale and Shapley [23])**.** *When all men and women have strict preferences there always exists a man-optimal stable marriage, and a woman-*

optimal stable marriage. Furthermore, the marriage produced by deferred acceptance algorithm with men proposing is a man-optimal stable marriage. The marriage produced by the algorithm with women as proposers is a woman-optimal stable marriage.

For the instance of SMrP given in Table 1.1, we obtain a marriage X_M by applying man-oriented algorithm `GS_Allocation` which is given by,

$$X_M = \{(m_1, w_1), (m_2, w_3), (m_3, w_4), (m_4, w_2)\}. \tag{1.3}$$

By applying the woman-oriented algorithm, we obtain X_W given by

$$X_W = \{(m_1, w_1), (m_2, w_3), (m_3, w_2), (m_4, w_4)\}. \tag{1.4}$$

The marriages X_M and X_W represent man-optimal and woman-optimal marriages, respectively.

After the publication of this paper [23], several other variants of the stable marriage problem have been discussed in the literature. The problem has provoked interest of people from many communities, especially from economists, game theorists and algorithmists. Also, there are a number of matching schemes in operation around the world which make use of some form of the `GS_Allocation`. In the United States of America, National Residents Matching Program (NRMP) [34], is a corporation to help in matching medical school students with residency positions. The NRMP has been using a form of the `GS_Allocation` since 1952, predating the publication of the paper about ten years. Each year, the NRMP conducts a match in which approximately 36,000 applicants compete for 25,000 available residency positions. The Canadian Resident Matching Service [9] and the Scottish Pre-Registration House Officer Allocations Scheme [44] are of great interest in this context.

1.2.3 The assignment game

In the marriage model due to Gale and Shapley [23] participants have strict preferences over the participants of the opposite set. The participants can not negotiate to make a partnership. These preferences do not include any exchange of money or goods. In other words, Gale and Shapley marriage model [23] studies the matching with non-transferable utilities. The simple reason for introducing transferable utilities in two-sided matching market is that many real life markets involve money. Modern game-theoretic concepts, when applied to even most elementary models, have often yielded suggestive results, sometimes reinforcing and sometimes challenging the more traditional doctrines based on behavioristic theories of the individual. A two-sided matching market in which money plays explicit role was presented by Shapley and Shubik [46] in 1972. In this model [46], each participant is either a seller or a buyer. Each seller has an indivisible commodity and each buyer has a certain amount of money. In such a market indivisible units of product like house or car etc are exchanged for money. The units need not be alike, and the same unit may have different values to different participants. The involvement of money in the model makes the assignment game different from marriage model. To explore the properties of such assignment game Shapley and Shubik concentrated on the core of the game. Before defining the core of a game first we review the definition of coalition.

Definition 1.18. Let $N = \{1, 2, \ldots, n\}$ be a set of participants of a game . A subset of N is called a *coalition*.

In a two-sided matching game the set of participants is divided in two disjoint sets say N_1 and N_2. Here $N_1 \times N_2$ denotes the set of all possible pairs in a two-sided matching game. A subset of $N_1 \times N_2$ is called a *coalition* for the two-sided matching game.

Definition 1.19. A *core* of the game is a set of outcomes upon which no coalition can receive a better profit or output from the game.

Definition 1.20. A *lattice* is a partially ordered set in which any two elements have a unique least upper bound and greatest lower bound. If all subsets of a lattice has a unique least upper bound and a unique greatest lower bound then the lattice is called a *complete lattice*.

Some of the main economic assumptions of the model [46] are:

- utility is defined with money,
- exchange of money is allowed,
- the objects of trade are indivisible,
- supply and demand functions are inflexible.

The second assumption is largely to avoid the necessity of providing by special rules for the ordinary payments from a customer to his supplier. The third assumption is rather unusual and is usually considered an imperfection in the market. Inflexibility of supply and demand functions means that these functions are step-functions[1] in nature. Here this assumption implies that each producer has a supply of exactly one item, and each consumer desires for exactly one item. This approach leads towards combinatorics and linear programming, and away from differential calculus methods. Though the assumptions are restrictive in many respects but there is no dearth of economic applications under this assumptions. In [46], Shapley and Shubik did not follow the algorithmic approach. They proved the existence of pairwise stable outcome by using the theory of linear programming, developed by Dantzig [10]. Now we discuss the assignment game.

[1]A step function is a special type of function whose graph is a series of line segments.

Let M and N be two disjoint sets where $|M| = m$ and $|N| = n$. Here M denotes the set of homeowners and N denotes the set of prospective purchasers. Homeowners and prospective purchasers are known as sellers and buyers, respectively. The seller i values his house at c_i, while the buyer j values the same house at h_{ij}. If seller i sells his house to buyer j for price p_i then profit or gain of seller i is

$$p_i - c_i$$

and buyer's gain is

$$h_{ij} - p_i.$$

The potential productivity for any pair (i, j) is denoted by a_{ij}. The potential productivity a_{ij} of two participants $i \in M$ and $j \in N$ is defined by

$$a_{ij} = v(\{i, j\}) = \max\left(0, h_{ij} - c_i\right), \tag{1.5}$$

where $v(\{i, j\})$ denotes the characteristic function for coalition of the participants i and j. More generally, a characteristic function for a coalition states the worth of the coalition. The characteristic function for a coalition S is denoted by $\nu(S)$. In two-side matching game no participant without the help of participant from the other set can effect a profitable transaction. Therefore, for two-sided assignment problem, the characteristic function satisfies the following condition:

$$\nu(S) = 0 \qquad \text{if } S \subseteq M \text{ or } S \subseteq N. \tag{1.6}$$

An assignment problem for larger mixed coalition is a selection of an optimal set of transactions, maximizing the coalition's total gain. Mathematically, the problem is to find

$$\nu(S) = \max\left[a_{i_1 j_1} + a_{i_2 j_2} + \ldots + a_{i_k j_k}\right] \tag{1.7}$$

$i_1, \ldots, i_k \in S \cap M$ and $j_1, \ldots, j_k \in S \cap N$ with $k = \min(|S \cap M|, |S \cap N|)$. The expression given by (1.7) is commonly called the *optimal assignment problem* for the assignment game.

For the solution of the assignment problem given by (1.7), Shapley and Shubik use the linear programming theory developed by Dantzig [10]. For such an assignment problem, mn non-negative real variables are introduced. These real variables are denoted by x_{ij}, for $i \in M$ and $j \in N$. To solve the assignment problem, $m + n$ constraints are imposed on x_{ij} as follows:

$$\sum_{i \in M} x_{ij} \leq 1, \qquad \sum_{j \in N} x_{ij} \leq 1. \tag{1.8}$$

Here, x_{ij} may be interpreted as probability that a seller i sells his house to a buyer j.

The linear programming (LP) problem is then to maximize the following objective function:

$$z = \sum_{i \in M} \sum_{j \in N} a_{ij} x_{ij}. \tag{1.9}$$

Dantzig [10] shows that the maximum value $z_{\max}$ is attained with all $x_{ij} \in \{0, 1\}$. Thus LP problem is equivalent to the assignment problem. Therefore, we have

$$z_{\max} = \nu(S),$$

where $S = M \cup N$.

The LP problem (1.9) with constraints given in (1.8) can be transposed into a dual form. In this case, the dual problem has $m + n$ non-negative real variables, $u_i, \ldots, u_m, v_1, \ldots, v_n$ subject to the mn constraints. For the dual problem, the objective function is to minimize

$$w = \sum_{i \in M} u_i + \sum_{j \in N} v_j \tag{1.10}$$

subject to the constraints

$$u_i + v_j \geq a_{ij} \qquad i \in M, j \in N. \tag{1.11}$$

Let $(u, v) = (u_1, \ldots, u_m, v_1, \ldots, v_n)$ be the vector which minimizes (1.10) subject to (1.11). Here, u_i and v_j denote the payoffs of seller i and buyer j, respectively. By (1.11), for every seller-buyer pair (i, j), we have

$$u_i + v_j \geq a_{ij} = \nu(\{i, j\}).$$

From (1.7), it follows that for all $i, j \in S$ we have

$$\sum_{i \in S \cap M} u_i + \sum_{j \in S \cap N} v_j \geq \sum_{i,j \in S} a_{ij}.$$

This can also be expressed as

$$\sum_{i \in S \cap M} u_i + \sum_{j \in S \cap N} v_j \geq \max \sum_{i,j \in S} a_{ij}.$$

By (1.7), we obtain

$$\sum_{i \in S \cap M} u_i + \sum_{j \in S \cap N} v_j \geq \nu(S). \tag{1.12}$$

Condition (1.12) ensures the non-improvability of (u, v). Now we use the fundamental duality theorem. This theorem states that

If feasible solutions to both the primal and dual systems exist, there exists an optimum solution to both systems and

$$z_{\max} = w_{\min}.$$

By using this, we can write

$$\sum_{i \in M} u_i + \sum_{j \in N} v_j = w_{\min} = z_{\max} = \nu(M \cup N) \tag{1.13}$$

which ensures the feasibility of (u, v). Thus both (1.13) and (1.12) guarantees that payoff vector (u, v) is feasible and not improvable by any coalition.

Theorem 1.21. *Shapley and Shubik [46] In core of an assignment game is precisely the set of solutions of the LP dual of the corresponding assignment problem.*

1.3 Generalizations of marriage model and assignment game

In [23], Gale and Shapley presented their model in which each participant has a strictly ordered preference list. The preferences of the participants in this model do not include money. The two main directions in which generalizations of the marriage model [23] have developed into literature involve models in which price setting is accomplished simultaneously with matching, and models involving matching with capacities. Shapely and Shubik [46] generalized the marriage model [23] by introducing money in matching markets. An interesting feature of these models is that quite similar results have been obtained for both these models [23] and [46], with fundamentally different proofs. The marriage model [23] and assignment game [46] show the non-emptiness of the set of stable matchings, using combinatorial arguments and linear programming arguments, respectively. Demange and Gale [11] presented a generalization of the assignment model in the spirit of [23]. In [41], Roth and Sotomayor have written a comprehensive survey of the theory of two-sided matching markets, mainly dealing with the marriage model and the assignment game. They raised the issue of explaining the similarities in behavior between the two models [23] and [46].

One approach towards unifying these models is to use linear programming formulations of stable matchings in discrete markets. Monographs [38, 43, 51] reveal surprising algebraic structure in the set of stable matchings of discrete markets, but a large difference between the linear programming formulations for the discrete case and the continuous case only emphasized that similar results were obtained for very different reasons in the two cases. Roth and Sotomayor [42] followed another direction. Their approach was to propose models which generalize both the marriage model [23] and the assignment game [46]. They presented a *generalized assignment game* that contains the both models [23] and [46] as a special case of their model [42]. Using this generalized assignment game [42], they tried to obtain the common results for the two in the more general model. But [42] can not guarantee that the solution set is non-empty. To fill this gap Eriksson and Karlander [14] introduced a model, the *RiFle assignment game*, where Ri stands for rigid and Fle stands for flexible. If participants in a matching model can negotiate to make a partnership then the participants are called *flexible*. On the other hand the participants which do not negotiate, like in marriage model, are called *rigid*. In order to prove the non-emptiness of the core and for lattice property of the core, Eriksson and Karlander [14] used an algorithmic approach. This approach, that uses the techniques introduced in [12], does not include the assignment game [46]. Using combinatorial arguments Sotomayor [48] gives a short prove of RiFle assignment game introduced by Eriksson and Karlander [14]. Recently, Farooq [15] presented a matching model. This model contains the models [23,42,46] as special cases. Farooq [15] also followed algorithmic approach to show that there always exist a stable matching for his model.

1.4 Basics of graph theory

Graph theory has a wide range of applications in engineering, social, physical, and biological sciences, linguistics, and in numerous other areas because of its inherent simplicity. To solve problems that arise in the fields of transportation and telecommunication, the operational research analysts often have to use different techniques. These techniques were first designed to solve classical problems such as the maximum flow problem, the independent set problem and the traveling salesman problem. Many of these problems are also closely related to the graph theory. Graph theory is extensively used in the study of discrete objects and their relationship in almost any physical situation. In this section we describe some basic definitions from graph theory which shall be used in subsequent chapters of this thesis.

1.4.1 Graphs, paths and cycles

Many real-world situations can be described by joining different points through lines. It is immaterial in such a drawing whether the lines are drawn straight or curved, long or short. A mathematical abstraction of such a drawing gives rise to the concept of a graph. In such drawing, the points may represent stations, electric devices, pieces of lands or persons and lines may represent roads connecting different stations, wire connection different electric devices, bridges connecting pieces of land or relationship between persons. As mentioned above the graph theory has wide range of applications. Perhaps this may be the reason that no unique terminology is followed for them in literature. These points and lines are named according to the situation to which the drawing is used. For example, the points may be called *vertices*, *nodes* or *junction points*; the lines may be called *arcs*, *edges*, *branches* or *links*. In this thesis we will use the following terminologies:

— *Vertices* for points.

— *Edges* for lines.

Now we define a graph formally.

Definition 1.22. A *graph* G is an ordered pair of a set of vertices V, and a set of edges E and is denoted by $G = (V, E)$.

Let e be an edge which connects a vertex v to a vertex v' in a graph G then the vertices v and v' are called the *end vertices* of the edge e.

Definition 1.23. Let $G = (V, E)$ be a graph and e be an edge with end vertices v and v' where e is directed from v to v' such an edge is called a *directed edge*.

A directed edge e with end vertices v and v' is denoted by an ordered pair (v, v'). An edge is called *undirected edge* if no direction is specified. An undirected edge with end vertices v and v' is represented by $\{v, v'\}$. In some graphs, the direction of edges is specified while in others the direction of edges is not specified. Therefore, there are two categories of graphs; directed and undirected.

Definition 1.24. A graph is called *directed* or *undirected*, depending upon the edges are directed or not.

Many definitions for undirected and directed graphs are same, although certain terms have slightly different meanings in the two contexts. For example, if v and v' are end vertices of an edge e, and the edge e is directed from the vertex v to the vertex v' then we say that the edge e is *incident from* the vertex v and is *incident to* the vertex v'. If v and v' are end vertices of an edge e in an undirected graph then we say that the edge e is *incident on* v and v'.

Definition 1.25. If end vertices of an edge are the same then it is called a *loop*.

The definition of parallel edges is slightly different in the case of directed and undirected graphs.

Definition 1.26. Let $G = (V, E)$ be an undirected graph. Let e and e' be two edges with same end vertices then e and e' are said to be *parallel edges*.

Definition 1.27. Let $G = (V, E)$ be a directed graph. Let e and e' be two edges edges in G such that both e and e' are incident from a vertex v to a vertex v' then e and e' are said to be *parallel edges*.

Definition 1.28. A graph without loops and parallel edges is called a *simple graph*.

Definition 1.29. In a graph $G = (V, E)$, a *walk* is a sequence $v_1, v_2, \ldots, v_k$ of k vertices such that each (v_j, v_{j+1}) for $1 \leq j \leq k-1$ is an edge in E.

Definition 1.30. A walk with no repetition of edges is called a *trail*.

Definition 1.31. A *path* is trail with all distinct vertices.

A path $S = v_1, v_2, \ldots, v_k$ is called a path from v_1 to v_k or a v_1-v_k path. Here v_1 and v_k are called *initial* and *terminal* vertices of the path S, respectively.

Definition 1.32. A *cycle* is a trail of the form of $v_1, v_2, \ldots, v_k, v_1$ with $k \geq 3$ such that $v_i \neq v_j$ for $1 \leq i < j \leq k$.

Definition 1.33. Let $G = (V, E)$ be a graph. If there exists a path from a vertex $v \in V$ to a vertex $v' \in V$ then the vertex v' is *reachable* from the vertex v.

Definition 1.34. Let $S = v_1, v_2, \ldots, v_k$ be a path from v_1 to v_k then

$$S_{ij} = v_i, v_{i+1}, \ldots, v_{j-1}, v_j$$

where $0 \leq i < j \leq k$ is called a *sub-path* of S from v_i to v_j.

1.4.2 Shortest path and its properties

Definition 1.35. For a graph $G = (V, E)$, we define a function $w : E \to \mathbb{R}$ which assigns a real number to each edge of the graph G. This function is known as *weight function*. The real number assigned by w to an edge is called *weight of the edge*.

Definition 1.36. A graph $G = (V, E)$ in which each edge has a weight is called a *weighted graph*.

A weight may represent distance, cost or current depending upon the nature of phenomenon under study. Several types of problems involving weighted graphs arise frequently in different fields like engineering, economics and optimization. Determining a shortest path between two vertices in a graph is one such problem.

Definition 1.37. In a weighted graph $G = (V, E)$, *weight of a walk* is the sum of the weights of the edges of the walk.

Similarly we can define the weights of paths and cycles.

Let $S = v_1, v_2, \ldots, v_k$ be a path in a weighted graph G then weight of S is given by

$$w(S) = \sum_{i=1}^{k-1} w(v_i, v_{i+1}).$$

The term shortest path is used in general sense and it represents a path having minimum weight.

Definition 1.38. Let $G = (V, E)$ be a weighted graph. A *shortest path* from a vertex v to a vertex v' is defined as any $v - v'$ path S with weight $w(S) = d(v, v')$ where $d(v, v')$ is given by

$$d(v, v') = \begin{cases} \min\{w(\widetilde{P}) \mid \widetilde{P} \text{ is a } v - v' \text{ path}\} & \text{if } v' \text{ is reachable from } v \\ +\infty & \text{otherewise.} \end{cases}$$

There are different variants of shortest path problem. However, in this thesis we consider *single-source shortest path problem.* A single-source shortest path problem states that for a given weighted graph $G = (V, E)$, find a shortest path from a given vertex v_0 to every other vertex $v \in V$. It is obvious from the definition of the weight function that weight of an edge is a real number which may be positive are negative. In literature, we find different algorithms to find a shortest path in a weighted graph. Some of these are designed to work for those graphs which have non-negative weights. For example, *Dijkstra's shortest path algorithm* [13] finds a shortest path from a given vertex v_0 to each vertex $v \in V$ in a weighted graph with non-negative weights.

Negative weights are not merely a mathematical curiosity; they arise in a natural way when we reduce other problems to shortest path problems.

Definition 1.39. A cycle C in a weighted graph with $w(C) < 0$ is called a *negative cycle*.

In a graph with negative weights there may exist a negative cycle and the Dijkstra's algorithm does not work in this case.

In solving a shortest path problem for graphs with negative weights, a negative cycle is very hard to deal with. If a graph contains a negative cycle which is reachable from the initial vertex v_0 then shortest path weights are not well defined. Because, otherwise, by traversing the negative weight cycle gives us a lesser weight path each time. However, if there is no such a cycle in the graph then shortest path weight remains well defined. The *Bellman-Ford algorithm* [6, 17] finds a shortest path for single-source shortest path problem for weighted graphs in which weights are real numbers.

Now we discuss some very useful properties of shortest path problem. One of the most important property is that a shortest path between two vertices contains other shortest path within it. This property of the shortest path is called *Optimal Substructure Property* of a shortest path. Shortest path algorithms typically exploit the optimal substructure property of a shortest path. This property is a hallmark of the applicability of both the dynamic programming and the greedy method which are useful tools for solving problems in mathematics and computer studies. The following lemma and its corollary state the property more precisely.

Lemma 1.40. *Cormen, Leiserson and Rivest [8] Given a graph* $G = (V, E)$ *with weight function* $w : E \to \mathbb{R}$. *Let* $S = v_1, v_2, \ldots, v_k$ *be a shortest path from a vertex* v_1 *to a vertex* v_k. *Let* $S_{ij} = v_i, v_{i+1}, \ldots, v_{j-1}, v_j$ *be a* $v_i - v_j$ *sub-path of the shortest path* S*, for any* i *and* j *such that* $1 \leq i < j \leq k$. *Then* S_{ij} *is a shortest path from* v_i *to* v_j.

A shortest path in a weighted graph can be decomposed into different sub-paths. These sub-paths exhibit a very interesting property that sum of the weights of these shortest sub-paths is same as the weight of the shortest path. We can rewrite this property as:

Corollary 1.41. *Cormen, Leiserson and Rivest [8] Let $G = (V, E)$ be a weighted graph with weight function $w : E \to \mathbb{R}$. Suppose a shortest path S from v_1 to v_k is decomposed in a $v_1 - v_{k-1}$ path S' and $S'' = v_{k-1}, v_k$. Then the weight of the shortest path S is given by*

$$d(v_1, v_k) = d(v_1, v_{k-1}) + d(v_{k-1}, v_k).$$

The following lemma describes that a shortest path from a given vertex v to any other vertex v' has a minimum weight among all paths from v to v'.

Lemma 1.42. *Cormen, Leiserson and Rivest [8] Let $G = (V, E)$ be a weighted graph with weight function $w : E \to \mathbb{R}$ and v be the initial vertex of a shortest path. Then, for all edges $(v', v'') \in E$, we have*

$$d(v, v'') \leq d(v, v') + w(v', v'').$$

1.4.3 Bipartite graphs

Bipartite graphs are very useful for modeling a phenomenon that involves relationships between two disjoint sets. It is extensively used in mathematical economics, computer sciences and modern coding theory. An example of bipartite graphs is a matching problem.

Definition 1.43. A graph $G = (V, E)$ is called *bipartite graph* if its vertex set V can be partitioned into two sets V_1 and V_2 such that each edge has one end in V_1 and the other in V_2. For such a bipartite graph, (V_1, V_2) is called a bipartition of V. In other words, no edge in a bipartite graph can have its end vertices from the same set.

Definition 1.44. A graph $G = (V, E)$ with bipartition (V_1, V_2) such that for any two vertices $v_1 \in V_1$ and $v_2 \in V_2$ there is an edge $(v_1, v_2) \in E$, is called a complete bipartite graph.

Graph theory is used for the study of matching theory as well. Here we review definitions to express the relationships between bipartite graphs and two-side matching problems.

Definition 1.45. Let $G = (V, E)$ be a graph. A *matching* $X \subseteq E$ is a collection of edges such that every vertex of V is incident to at most one edge of X.

If the graph is a bipartite graph then the matching is called a *bipartite matching*. A two-side matching is in fact a bipartite matching.

Definition 1.46. Let $G = (V, E)$ be a graph. If a vertex v is not an end vertex of any edge of a matching X then v is said to be *unmatched* in X.

By *size* of a matching we mean the number of edges in the matching.

Definition 1.47. A *maximum matching* in a graph is a matching of maximum size among all matchings in the graph.

Definition 1.48. A *maximal matching* in a graph is a matching that cannot be enlarged by adding an edge.

A maximal matching X has the property that if any edge not in X is added to X, it is no longer a matching, that is, X is maximal if it is not a proper subset of any other matching in graph. Note that every maximum matching is maximal, but every maximal matching may not be a maximum matching.

Definition 1.49. A *perfect matching* is a matching which matches all vertices of the graph. That is, every vertex of the graph is an end vertex of exactly one edge of the matching.

The marriages given by (1.3) and (1.4) represent perfect bipartite matchings. Every perfect matching is maximum and hence, maximal. In some literature, the term complete matching is used instead of perfect matching.

Weighted bipartite graphs give rise to the idea of weighted bipartite matchings.

Definition 1.50. Let $G = (V, E)$ be a weighted graph and X be a matching in G then the *weight of matching* X is the sum of the weights of edges in X, that is,

$$w(X) = \sum_{e \in X} w(e).$$

Definition 1.51. A *maximum weighted matching* is defined as a matching such that the sum of the weights of the edges in the matching is maximum.

There are different methods available for obtaining the maximum weighted bipartite matchings. For a detailed study of these methods, readers are referred to West [52] and Schrijver [47].

Chapter 2

Job market with linear valuations and possibly bounded salaries

2.1 Introduction

The motivation of this chapter is to prove the existence of pairwise stable outcome for a two-sided one-to-many matching market. In a one-to-many matching market, participants on one side can match up with at most one participant of the opposite side, whereas the participants on the opposite side are allowed to make partnerships with as many participants as they want to. There are many examples of such a model in real life. For example, consider a job market in which workers are restricted to work in at most one firm, whereas the firms are free to hire more than one workers depending upon their capacities. Along with marriage model, a one-to-many matching model is also discussed by Gale and Shapley in their paper [23]. The one-to-many model by Gale and Shapley [23] is also known as *college admission problem*. In their model, each student seeks for admission in a college. On the other hand, each college wants to enroll students to fulfil all available seats in the college. The available seats or the maximum number of students which a college can enroll is called *quota* of the college. Since a student can join only one college, therefore, quota of each student is one. In this chapter, we consider a similar type of two-sided matching model, that is, a one-to-many matching model. Preferences of participants in our model are similar to those which were considered in the model of Farooq [15]. In [15], Farooq developed a procedure to show that there always exist a pairwise stable outcome for one-to-one matching model with linear valuations. Motivated by his work we consider a model which is general than the model given by Farooq [15]. Our procedure to find a pairwise stable outcome is some what similar to the procedure developed by Farooq [15]. In the theory of stable matchings, generally it is believed that when one develops an algorithm to show the existence of a stable matching for the one-to-one models (the marriage markets), the same ideas may be adapted to show the existence of a stable matching for one-to-many models (the college admissions markets). However, due to the generality of our model, the simple adaptation of the old algorithm for one-to-one model, given by Farooq [15], does not work in our case. A comprehensive work is done in this chapter to extend the ideas of the algorithm for one-to-one model (Farooq [15]) to the algorithm for one-to-many model. The marriage model [23], assignment game [46] and hybrid models [14, 15, 48] are some special cases of our model.

We organize this chapter as follows. In Section 2.2, we describe important features of our model. Also we introduce some basic terminologies and notations for our model. We define blocking pair and pairwise stability for this model in Section 2.3. The characterization theorem for the pairwise stability is proved in Section 2.4. Section 2.5 is devoted for the comparison of our model with some well known existing models. In Section 2.6, we present an algorithm to find a pairwise stable outcome for our model. Correctness and analysis of the algorithm are shown in the same section.

2.2 Model description

We consider a job market in which the set of participants is partitioned into two disjoint sets; a set of individuals and a set of institutions. Generically, we recognize the individuals as workers and the institutions as firms. The basic problem in such a market is to assign the workers and firms to each other. Here we list some important features of our model:

- the set of participants is partitioned into two sets; the set of firms and set of workers,
- a worker can work for at most one firm,
- each firm has a certain quota to employ workers,
- salaries are allowed,
- each worker-firm pair may have lower and upper bounds on the salary,
- the preferences of the participants are identified by strictly increasing and linear functions,
- strictly increasing linear functions are known as linear valuations.

Valuations can be defined in different ways. Here by valuation, we mean estimation of the value of some asset or real property. In this model, workers of a firm provide their service to increase the revenue of the firm. As a result of these services, a certain amount of money is paid by firm to a worker. This is known as *salary* of a worker. We remark that the restrictions of boundedness or unboundedness on the salaries do not impact the termination of the algorithm since preferences of the participants in our model are represented by strictly increasing and linear valuations. For example, in unrestricted case, the firm's individual rationality constraint induces upper bound and the worker's individual rationality constraint induces lower bound for the salaries. Therefore, the possible bounded salaries can not be considered as novelty in our model. Rather, it would help us to understand the comparison of our model and the known models. Now we describe our model mathematically.

We consider two finite disjoint sets of participants P and Q where P is the set of workers and Q is the set of firms. The set of all possible worker-firm pairs is denoted by E, that is,

$$E = \{(i, j) \mid i \in P \text{ and } j \in Q\}.$$

We assume that each worker can work for at most one firm, that is, quota of each worker is one. Each firm can employ as many workers as it wishes. For each $j \in Q$, $\mu(j)$ is called the quota of firm j. It denotes the maximum number of workers j can employ. Here $\mu = (\mu(j) \mid j \in Q) \in \mathbb{Z}_+^Q$, where $\mathbb{Z}_+^Q$ is the set of positive vectors of $\mathbb{Z}^Q$. If the number of workers hired by a firm j are less than $\mu(j)$ then the firm j is called *unsaturated*. If workers hires by a firm j is the same as $\mu(j)$ then the firm j is called *saturated*.

Assume that each worker-firm pair (i, j) may have lower and upper bounds on the salary, that is, the salaries are possibly bounded. The lower and upper bounds on the salaries are expressed by two vectors $\underline{\pi}$ and $\overline{\pi}$, where $\underline{\pi} \in (\mathbb{R} \cup \{-\infty\})^E$, $\overline{\pi} \in (\mathbb{R} \cup \{+\infty\})^E$ and $\underline{\pi} \leq \overline{\pi}$. This means that for any two vectors $\underline{\pi} \in (\mathbb{R} \cup \{-\infty\})^E$ and $\overline{\pi} \in (\mathbb{R} \cup \{+\infty\})^E$, $\underline{\pi}_{ij} \leq \overline{\pi}_{ij}$ for all $(i, j) \in E$. A vector $s = (s_{ij} \mid (i, j) \in E) \in \mathbb{R}^E$ is called a *feasible salary vector* if $\underline{\pi} \leq s \leq \overline{\pi}$.

We also assume that each worker has a list of preferences of those firms where he/she is willing to work. Similarly, each firm has preferences over those workers whom the firm wants to employ. The preferences of the participants are represented by continuous, strictly increasing linear functions, which are called linear valuations in our work. For each $(i, j) \in E$, $\nu_{ij} : \mathbb{R} \to \mathbb{R}$ represents the valuation of a worker i for a monetary transfer from a firm j to i. Similarly, $\nu_{ji} : \mathbb{R} \to \mathbb{R}$ represents the valuation of the firm j for a monetary transfer from a worker i to j.

The monetary transfer from a worker to a firm should not be surprising. For example, a worker can agree on the reduction of his/her demanded salary after negotiation with a firm.

We say that a firm j is *acceptable* to a worker i at $\alpha \in \mathbb{R}$ if $\nu_{ij}(\alpha) \geq 0$. Similarly, a worker i is *acceptable* to a firm j at $\alpha \in \mathbb{R}$ if $\nu_{ji}(\alpha) \geq 0$. If $\nu_{ij}(\alpha) \geq 0$ and $\nu_{ji}(-\alpha) \geq 0$ then the pair (i, j) is called *mutually acceptable.*

A worker i *prefers* a firm j to a firm j' at $\alpha, \alpha' \in \mathbb{R}$ if $\nu_{ij}(\alpha) > \nu_{ij'}(\alpha')$ and i is *indifferent* between j and j' at $\alpha, \alpha' \in \mathbb{R}$ if $\nu_{ij}(\alpha) = \nu_{ij'}(\alpha')$. Similarly, a firm j *prefers* a worker i to a worker i' at $\alpha, \alpha' \in \mathbb{R}$ if $\nu_{ji}(\alpha) > \nu_{ji'}(\alpha')$ and j is *indifferent* between i and i' at $\alpha, \alpha' \in \mathbb{R}$ if $\nu_{ji}(\alpha) = \nu_{ji'}(\alpha')$.

A set $X = \{(S_j, j) \mid j \in Q\} \subseteq 2^P \times Q$ is called a *job allocation* if

(i) $|S_j| \leq \mu(j)$ for all $j \in Q$.

(ii) $S_j \cap S_{j'} = \emptyset$ for all $j, j' \in Q$ with $j \neq j'$.

Here S_j denotes the set of workers hired by a firm j. Condition (i) says that in a job allocation workers hired by a firm j must not exceed the quota of j. Condition (ii) reads that if a worker is working for a firm j then he can not work for any other firm j'.

In the sequel, whenever we say that $S_j \in X$ (or $j \in X$), we always mean that $(S_j, j) \in X$. For any $j \in Q$, we reserve the notation S_j for X only.

We say that a firm j *employs* a worker i if $i \in S_j$. Obviously, a firm j is *businessless* if $S_j = \emptyset$. Similarly, a worker i is said to be *unemployed* if $i \notin S_j$ for all $j \in Q$. We set $S_i = \{j\}$ if $i \in S_j$ and $S_i = \emptyset$ if i is unemployed. We say that a worker i and a firm j are *matched* if $i \in S_j$.

2.3 Blocking pair and pairwise stable outcome

A quadruple $(X; s, q, r)$ is said to be an *outcome* if X is a job allocation, s is a feasible salary vector and $(q, r) \in \mathbb{R}^P \times \mathbb{R}^Q$ is defined by

$$q_i = \begin{cases} \nu_{ij}(s_{ij}) & \text{if } i \in S_j \text{ for some } j \in Q \\ 0 & \text{otherwise} \end{cases} \quad (\forall i \in P), \tag{2.1}$$

$$r_j = \begin{cases} \min\{\nu_{ji}(-s_{ij}) \mid i \in S_j\} & \text{if } |S_j| = \mu(j) \\ 0 & \text{otherwise} \end{cases} \quad (\forall j \in Q), \tag{2.2}$$

where the minimum over an empty set is defined to be 0.

An outcome $(X; s, q, r)$ is *blocked* by a worker-firm pair (i, j) if

$$i \notin S_j \text{ and } \nu_{ij}(s_{ij}) > q_i, \quad \nu_{ji}(-s_{ij}) > r_j. \tag{2.3}$$

The statement in (2.3) is equivalent to saying that i and j are not matched but i prefers j to his/her current employer[1] and j prefers i at least to one of his worker or still have a vacancy to employ i.

An outcome $(X; s, q, r)$ is *pairwise stable* if the following two conditions are satisfied:

(ps1) $\nu_{ij}(s_{ij}) \geq 0$ and $\nu_{ji}(-s_{ij}) \geq 0$ for all $(i, j) \in E$ with $i \in S_j$.

(ps2) $\nu_{ij}(\alpha) \leq q_i$ or $\nu_{ji}(-\alpha) \leq r_j$ for all $\alpha \in \mathbb{R}$ with $\underline{\pi}_{ij} \leq \alpha \leq \overline{\pi}_{ij}$ and for all $(i, j) \in E$ with $i \notin S_j$.

Condition (ps1)[2] says that if a firm employs a worker then both are acceptable to each other. Condition (ps2) means $(X; s, q, r)$ is not blocked by any worker-firm pair.

A job allocation X is called *pairwise stable* if $(X; s, q, r)$ is pairwise stable.

[1]For convenience, we say that a worker is self-employed if he/she is unemployed.

[2]ps stands for pairwise stability.

2.4 Characterization of a pairwise stable outcome

This section is devoted to the characterization of a pairwise stable outcome. We shall use this characterization to devise the algorithm in Section 2.6.

A *characteristic vector* $\chi_S \in \{0,1\}^P$ of a set $S \subseteq P$ is defined by

$$\chi_S(k) = \begin{cases} 1 & \text{if } k \in S, \\ 0 & \text{if } k \in P \setminus S. \end{cases}$$

Analogously, we can define a characteristic vector $\chi_S \in \{0,1\}^Q$ of a set $S \subseteq Q$. Obviously, χ_S is the zero vector if $S = \emptyset$. Next theorem gives a characterization of a pairwise stable outcome. This theorem is a modification of Theorem 2.1 [16] related to our model.

Theorem 2.1. *Let X be a job allocation. There exists a feasible salary vector s and a vector (q,r) defined by (2.1) and (2.2) forming a pairwise stable outcome $(X;s,q,r)$ if and only if there exists a feasible salary vector p and vectors $z_P, z_Q \in \{0,1\}^E$ such that*

(ps′1) *for all $i \in P$, χ_{S_i} is an optimal solution of*

$$\begin{array}{ll} \text{maximize}_{k\in Q} & \nu_{ik}(p_{ik})\chi_S(k) \\ \text{subject to} & S \subseteq Q, \\ & |S| \leq 1, \\ & \chi_S(k) \leq z_P(i,k) \quad (k \in Q). \end{array} \tag{2.4}$$

(ps′2) *for all $j \in Q$, χ_{S_j} is an optimal solution of*

$$\begin{array}{ll} \text{maximize} & \sum_{k\in P} \nu_{jk}(-p_{kj})\chi_S(k) \\ \text{subject to} & S \subseteq P, \\ & |S| \leq \mu(j), \\ & \chi_S(k) \leq z_Q(k,j) \quad (k \in P). \end{array}$$

(ps′3) $z_P \vee z_Q = \mathbf{1}$.[3]

(ps′4) $z_P(i,j) = 0 \Rightarrow p_{ij} = \underline{\pi}_{ij}$ *and* $z_Q(i,j) = 0 \Rightarrow p_{ij} = \overline{\pi}_{ij}$.

Proof. (⇐) Suppose that there exist a feasible salary vector p, and z_P, $z_Q \in \{0,1\}^E$ such that the conditions (ps′1)−(ps′4) are satisfied. We prove that $(X; s, q, r)$ satisfies (ps1) and (ps2), where $s = p$ and (q, r) is defined by (2.1) and (2.2). The condition (ps1) is implied by (ps′1) and (ps′2). We only show that (ps2) is also satisfied.

Suppose that (ps2) does not hold. Then there exists $(i,j) \in E$ with $i \notin S_j$ and $\alpha \in \mathbb{R}$ with $\underline{\pi}_{ij} \leq \alpha \leq \overline{\pi}_{ij}$ such that $\nu_{ij}(\alpha) > q_i$ and $\nu_{ji}(-\alpha) > r_j$. If $\nu_{ij}(\alpha) > q_i$ then by (ps′1) and the fact that ν_{ij} is increasing, at least one of the following two cases must hold:

(a1) $z_P(i,j) = 0$ or (a2) $z_P(i,j) = 1$ and $p_{ij} < \alpha$.

Similarly, if $\nu_{ji}(-\alpha) > r_j$ then by the fact that ν_{ji} is increasing, at least one of the following two cases must hold:

(b1) $z_Q(i,j) = 0$ or (b2) $z_Q(i,j) = 1$ and $p_{ij} > \alpha$.

Obviously, (a1) and (b1) can not hold together by (ps′3). If (a1) and (b2) are true then (ps′4) yields that $p_{ij} = \underline{\pi}_{ij}$ which is not possible. With the same argument, (a2) and (b1) can not be true together. The statements (a2) and (b2) are obviously incompatible. Therefore (ps2) must hold.

(⇒) Suppose that there exists a feasible salary vector s such that $(X; s, q, r)$ is a pairwise stable outcome, where (q, r) is defined by (2.1) and (2.2). Then (ps1) and (ps2) hold true. We prove that there exist p and $z_P, z_Q \in \{0,1\}^E$ which

[3] For any $(i,j) \in E$, the (i,j)-th component of $z_P \vee z_Q$ is given by $(z_P \vee z_Q)(i,j) = \max\{z_P(i,j), z_Q(i,j)\}$.

satisfy (ps′1)−(ps′4). Define z_P and z_Q as follows:

$$z_P(i,j) = \begin{cases} 0 & \text{if } \nu_{ij}(\underline{\pi}_{ij}) > q_i \text{ and } i \notin S_j \\ 1 & \text{otherwise} \end{cases} \qquad (\forall (i,j) \in E),$$

$$z_Q(i,j) = \begin{cases} 0 & \text{if } \nu_{ji}(-\overline{\pi}_{ij}) > r_j \text{ and } i \notin S_j \\ 1 & \text{otherwise} \end{cases} \qquad (\forall (i,j) \in E).$$

From the definitions of z_P and z_Q, we observe that for any $(i,j) \in E$ with $i \notin S_j$, the following holds:

$$z_P(i,j) = z_Q(i,j) = 1 \Rightarrow \nu_{ij}(\underline{\pi}_{ij}) \leq q_i \text{ and } \nu_{ji}(-\overline{\pi}_{ij}) \leq r_j. \tag{2.5}$$

The statement (2.5) together with (ps2) implies that

$$\exists \gamma_{ij} \in [\underline{\pi}_{ij}, \overline{\pi}_{ij}] \text{ such that } \nu_{ij}(\gamma_{ij}) \leq q_i \text{ and } \nu_{ji}(-\gamma_{ij}) \leq r_j \tag{2.6}$$
$$\text{for all } (i,j) \in E \text{ with } z_P(i,j) = z_Q(i,j) = 1 \text{ and } i \notin S_j.$$

Now define the salary vector $p \in \mathbb{R}^E$ as follows:

$$p_{ij} = \begin{cases} \underline{\pi}_{ij} & \text{if } z_P(i,j) = 0 \\ \overline{\pi}_{ij} & \text{if } z_Q(i,j) = 0 \\ \gamma_{ij} & \text{if } z_P(i,j) = z_Q(i,j) = 1 \text{ and } i \notin S_j \\ s_{ij} & \text{otherwise} \end{cases} \qquad (\forall (i,j) \in E), \tag{2.7}$$

where γ_{ij}, for $(i,j) \in E$ with $z_P(i,j) = z_Q(i,j) = 1$ and $i \notin S_j$, is defined by (2.6). The definitions of z_P and z_Q along with (ps2) imply (ps′3) and the definition of p implies (ps′4). Next we prove (ps′2). Let $j \in Q$ and $S \subseteq P$ is such that $|S| \leq \mu(j)$ and $\chi_S(k) \leq z_Q(k,j)$ for all $k \in P$. For any $i \in S \setminus S_j$, it is enough to show that

$$\nu_{ji}(-p_{ij}) \leq r_j. \tag{2.8}$$

Since $i \in S$ we must have $z_Q(i,j) = 1$. If $z_P(i,j) = 1$ then (2.6) and the definition of p implies (2.8). If $z_P(i,j) = 0$ then the definitions of z_P and p give $\nu_{ij}(p_{ij}) > q_i$. This along with (ps2) gives (2.8). Analogously, we can prove (ps′1). This completes the proof. □

2.5 Comparison of our model with existing models

In this section, we compare our model with few existing models that are directly related to our model. With each $(i, j) \in E$, we associate a pair (β_{ij}, β_{ji}) of real numbers. We say that $j \in Q$ is *acceptable* to $i \in P$ if $\beta_{ij} \geq 0$. Similarly, i is *acceptable* to j if $\beta_{ji} \geq 0$. A (one-to-one) matching is a subset of E such that each participant appears at most once.

Marriage model

In the marriage model, P and Q are recognized as a set of men and a set of women, respectively. Negotiations and salaries are not permitted in this model. That is why, the participants in the marriage model are called "rigid". Associated with each participant, there is a strictly ordered finite list containing each participant of the opposite side, that is, all participants are mutually acceptable. A stable matching is a one-one pairing of acceptable men and women such that there are no man and woman who are not a pair but prefer each other to their assigned partners.

Next we present the model mathematically as described by Fujishige and Tamura [19]. The numbers β_{ij} and β_{ji} denote the preferences of i and j, respectively. We say that a man i *prefers* a woman j_1 to a woman j_2 if $\beta_{ij_1} > \beta_{ij_2}$. Analogously, a woman j *prefers* a man i_1 to a man i_2 if $\beta_{ji_1} > \beta_{ji_2}$. A matching X is called pairwise stable if there exists $(q, r) \in \mathbb{R}^P \times \mathbb{R}^Q$ such that

(m1) $q_i = \beta_{ij}$ and $r_j = \beta_{ji}$ for all $(i, j) \in X$.

(m2) $q \geq \mathbf{0}$, $r \geq \mathbf{0}$ and $q_i = 0$ (resp. $r_j = 0$) if i (resp. j) is unmatched.

(m3) $q_i \geq \beta_{ij}$ or $r_j \geq \beta_{ji}$ for all $(i, j) \in E$.

In the pioneering work by Gale and Shapley [23] in two sided matching markets, an algorithm is proposed which produces a stable matching. The main characteristic of their algorithm is that when men are the proposer, it gives a unique stable matching and each man has the best partner that he can have in any other stable matching. However, if the role of the sexes are reversed, the algorithm yields a unique stable matching where each woman has the best partner that she can have in any other stable matching.

Assignment game

In the one-to-one buyer-seller model by Shapley and Shubik [46], known as assignment game, they proved non constructively the existence of a stable outcome and showed that the set of stable outcomes and the core of the game are the same. In contrast to the marriage model [23], money is involved in the assignment game and each participant can negotiate with the participants of the opposite side. This freedom makes the assignment game [46] different from the marriage model [23]. The participants of the assignment game [46] are called "flexible".

To describe mathematically, we associate a number $\beta_{ij} + \beta_{ji}$ with each pair $(i, j) \in E$. The number $\beta_{ij} + \beta_{ji}$ is called the joint productivity or the worth of the pair (i, j) if they form a partnership (Note that we could have denoted the joint productivity by a single number γ_{ij}, say, but we write it as a sum of two numbers to compare the assignment game with other models). A matching X is called pairwise stable if there exists $(q, r) \in \mathbb{R}^P \times \mathbb{R}^Q$ such that

(a1) $\sum_{i \in P} q_i + \sum_{j \in Q} r_j = \sum_{(i,j) \in X} (\beta_{ij} + \beta_{ji})$.

(a2) $q \geq \mathbf{0}$ and $\mathbf{r} \geq \mathbf{0}$.

(a3) $q_i + r_j \geq \beta_{ij} + \beta_{ji}$ for all $(i, j) \in E$.

An easy implication of (a1)−(a3) is that $q_i + r_j = \beta_{ij} + \beta_{ji}$ for all $(i, j) \in X$ and $q_i = 0$ (resp. $r_j = 0$) if i (resp. j) is unmatched. In the marriage model, (m1) means that the participants always demand the prescribed payoffs. However, in the assignment game the equality $q_i + r_j = \beta_{ij} + \beta_{ji}$ reflects that the participants can negotiate and distribute the amount $\beta_{ij} + \beta_{ji}$ between each other. For the above model, Shapley and Shubik [46] proved non constructively the existence of a stable outcome and showed that the set of stable outcomes and the core of the game are the same.

Mixed market model

We have discussed that the marriage model [23] consists of rigid participants only where as the assignment game [46] consists of flexible participants only. It is very natural to think of a single market which consists of both rigid and flexible participants. Eriksson and Karlander [14] proposed a mixed market model (the RiFle assignment game) consisting of both the rigid participants and the flexible participants. They proved the existence of stable outcome. However, their proofs do not hold for all continuous markets (the assignment game due to Shapley and Shubik [46]). Sotomayor [48] also considered the mixed market model that contains both the marriage model and the assignment game as special cases. Her model is a generalization of the RiFle assignment game in the sense that her proofs hold for both discrete and continuous markets. Mainly using her terminologies, we describe here this hybrid model mathematically.

We think of the sets P and Q as sets of workers and firms, respectively. We partition the participants into two classes R and F where R is the set of rigid participants and F is the set of flexible participants. Define two subsets R^* and

F^* of E by:

$$
\begin{aligned}
R^* &= \{(i,j) \in E \mid i \in R \text{ or } j \in R\}, \\
F^* &= \{(i,j) \in E \mid i,j \in F\}.
\end{aligned}
$$

R^* and F^* are called the sets of rigid and flexible pairs, respectively. A matching X is called pairwise stable if there exists $(q,r) \in \mathbb{R}^P \times \mathbb{R}^Q$ such that

(h1) $q_i + r_j = \beta_{ij} + \beta_{ji}$ for all $(i,j) \in X$.

(h2) $q_i = \beta_{ij}$ and $r_j = \beta_{ji}$ for all $(i,j) \in X \cap R^*$.

(h3) $q \geq \mathbf{0}$, $r \geq \mathbf{0}$ and $q_i = 0$ (resp. $r_j = 0$) if i (resp. j) is unmatched.

(h4) $q_i + r_j \geq \beta_{ij} + \beta_{ji}$ for all $(i,j) \in F^*$.

(h5) $q_i \geq \beta_{ji}$ or $r_j \geq \beta_{ji}$ if $(i,j) \in R^*$.

Sotomayor [48] proved the existence of the stable outcome in this model. Further she proved that the core is a complete lattice. One can easily see that if $F^* = \emptyset$ then the above model coincides with the marriage model by Gale and Shapley [23]. Also, if $R^* = \emptyset$ then it would coincide with the assignment game by Shapley and Shubik [46].

Let us assume that $\mu = (1, \ldots, 1) \in \mathbb{Z}_+^Q$ and define the linear valuations in a special way as follows:

$$
\nu_{ij}(s_{ij}) = \beta_{ij} + s_{ij}, \quad \nu_{ji}(-s_{ij}) = \beta_{ji} - s_{ij} \qquad (\forall (i,j) \in E),
$$

where $s_{ij} \in \mathbb{R}$ with $\underline{\pi}_{ij} \leq s_{ij} \leq \overline{\pi}_{ij}$ and $\beta_{ij}, \beta_{ji} \in \mathbb{R}$. If we fix $\underline{\pi} = \overline{\pi} = \mathbf{0}$ then the marriage model due to Gale and Shapley [23] becomes a special case of our model. If we let $\underline{\pi} = (-\infty, \ldots, -\infty)$ and $\overline{\pi} = (+\infty, \ldots, +\infty)$ then we get the assignment game due to Shapley and Shubik [46]. Now we assume that $\underline{\pi} \in (\mathbb{R} \cup \{-\infty\})^E$ and $\overline{\pi} \in (\mathbb{R} \cup \{+\infty\})^E$, that is, the set of pairs is partitioned randomly in to the set of rigid pairs and flexible pairs. This then shows that the hybrid models of Eriksson and Karlander [14] and Sotomayor [48] are special cases of our model.

2.6 An algorithm for finding a stable job allocation

This section deals with finding a stable job allocation for our model described in Section 3.2. We do this by establishing an algorithm which is an extended version of the algorithm proposed by Farooq [15]. His algorithm works when each worker can work for at most one firm and each firm can employ at most one worker, and the salaries are bounded. The algorithm proposed in this section includes the cases where firms can employ as many workers as they wish and when the salaries have no bounds. At the end of this section, we show that the algorithm works correctly and terminates after a finite number of iterations.

Let us define the valuations ν_{ij} and ν_{ji} as follows:

$$\nu_{ij}(x) = \alpha_{ij}x + \beta_{ij}, \quad \nu_{ji}(x) = \alpha_{ji}x + \beta_{ji} \qquad (\forall (i,j) \in E), \tag{2.9}$$

where α_{ij} and α_{ji} are given positive real numbers and, β_{ij} and β_{ji} are any given real numbers.

Initially, we define the salary vector $p \in \mathbb{R}^E$ by

$$p_{ij} := \begin{cases} \overline{\pi}_{ij} & \text{if } \nu_{ji}(-\overline{\pi}_{ij}) \geq 0 \text{ and } \overline{\pi}_{ij} < +\infty \\ \max\{\underline{\pi}_{ij}, \frac{\beta_{ji}}{\alpha_{ji}}\} & \text{otherwise} \end{cases} \qquad (\forall (i,j) \in E). \tag{2.10}$$

Then $\underline{\pi}_{ij} \leq \max\{\underline{\pi}_{ij}, \frac{\beta_{ji}}{\alpha_{ji}}\} \leq p_{ij} \leq \overline{\pi}_{ij}$ for all $(i,j) \in E$. That is, p is a feasible salary vector. Now define $z_P \in \{0,1\}^E$ as follows:

$$z_P(i,j) = \begin{cases} 1 & \text{if } \nu_{ji}(-p_{ij}) \geq 0 \\ 0 & \text{otherwise} \end{cases} \qquad (\forall (i,j) \in E). \tag{2.11}$$

We next define $z_0 \in \{0,1\}^E$ by

$$z_0(i,j) = \begin{cases} 0 & \text{if } \nu_{ij}(p_{ij}) \leq 0 \\ 1 & \text{otherwise} \end{cases} \qquad (\forall (i,j) \in E), \tag{2.12}$$

and fix

$$\tilde{z}_P = z_P \wedge z_0.^4 \tag{2.13}$$

Note that any $(i,j) \in E$ with $\tilde{z}_P(i,j) = 1$ implies that $\nu_{ij}(p_{ij}) > 0$ and $\nu_{ji}(-p_{ij}) \geq 0$, that is, i and j are mutually acceptable. Furthermore, define $\hat{z}_P \in \{0,1\}^E$ by

$$\hat{z}_P(i,j) = \begin{cases} 1 & \text{if } \tilde{z}_P(i,j) = 1 \text{ and } \nu_{ij}(p_{ij}) = \\ & \max\{\nu_{ij'}(p_{ij'}) \mid j' \in Q, \tilde{z}_P(i,j') = 1\} \\ 0 & \text{otherwise} \end{cases} \qquad (\forall (i,j) \in E). \tag{2.14}$$

Thus for any $(i,j) \in E$ with $\hat{z}_P(i,j) = 1$, we see that the worker i and the firm j are mutually acceptable and i prefers j to all those firms which accept i. Also for any $j \in Q$ and $S \in 2^Q$, define $r_j^S \in \mathbb{R}$ by

$$r_j^S = \min\{\nu_{ji}(-p_{ij}) \mid i \in S\}. \tag{2.15}$$

Let $S \in 2^P$, $z_Q \in \{0,1\}^E$ and p be a feasible salary vector. Then the 3-tuple (S, z_Q, p) is said to be a *best choice* for $j \in Q$ if it satisfies the following property:

(BC) $\forall i \in P \setminus S$ with $z_Q(i,j) = 1$, the following hold:

(a) $\nu_{ji}(-p_{ij}) \leq r_j^S$.

(b) $|S| \leq \mu(j)$ and if $\nu_{ji}(-p_{ij}) > 0$ then $|S| = \mu(j)$.

To find a matching, initially we define a vector $z_Q \in \{0,1\}^E$ and a vector $\tilde{\mu} \in \mathbb{Z}_+^Q$ as follows:

$$z_Q(i,j) := \begin{cases} 1 & \text{if } p_{ij} < \overline{\pi}_{ij} \text{ or} \\ & [\overline{\pi}_{ij} = \underline{\pi}_{ij} \text{ and } \nu_{ji}(-\overline{\pi}_{ij}) < 0] \\ 0 & \text{otherwise} \end{cases} \qquad (\forall (i,j) \in E), \tag{2.16}$$

[4] For any $(i,j) \in E$, the (i,j)-th component of $z_P \wedge z_0$ is given by $(z_P \wedge z_0)(i,j) = \min\{z_P(i,j), z_0(i,j)\}$.

$$\tilde{\mu} = (\tilde{\mu}(j) = 0 \mid j \in Q). \tag{2.17}$$

Now, for each $j \in Q$, define η_j by

$$\begin{aligned}\eta_j = \{S \in 2^P \mid (S, z_Q, p) &\text{ satisfies (BC) for } j, \tilde{\mu}(j) \leq |S| \\ &\text{and } \chi_S(i) \leq \hat{z}_P(i,j) \quad (\forall i \in P)\}\end{aligned} \tag{2.18}$$

and let

$$\eta = \cup_{j \in Q}\{\eta_j\}. \tag{2.19}$$

We further define a set Γ as follows:

$$\Gamma = \cup_{j \in Q}\{\eta_j \times \{j\}\}. \tag{2.20}$$

Find a matching $X = \{(S_j, j) \mid j \in Q\}$[5] in the bipartite graph $(\eta, Q; \Gamma)$ which satisfies the following:

$$S_j \cap S_{j'} = \emptyset \text{ for all } S_j, S_{j'} \in X \text{ with } j \neq j', \tag{2.21}$$

$$\sum_{(i,j)\in E} \nu_{ji}(-p_{ij})\chi_{S_j}(i) \text{ is maximum among the matchings satisfying (2.21),} \tag{2.22}$$

$$\sum_{(i,j)\in E} (\ln \alpha_{ji} - \ln \alpha_{ij})\chi_{S_j}(i) \text{ is maximum among the matchings satisfying (2.21) and (2.22).} \tag{2.23}$$

Then obviously X is a job allocation. Let S_P be the set of all workers which are employed by some firm, that is,

$$S_P = \cup_{j \in Q} S_j. \tag{2.24}$$

We redefine $z_Q \in \{0,1\}^E$ and $\tilde{\mu} \in \mathbb{Z}_+^Q$ by

$$z_Q(i,j) := \begin{cases} 1 & \text{if } p_{ij} < \overline{\pi}_{ij} \text{ or } i \in S_j \\ z_Q(i,j) & \text{otherwise} \end{cases} \qquad (\forall (i,j) \in E), \tag{2.25}$$

[5]For the sake of convenience, X is represented in this form. For instance, if some $j \in Q$ is not matched in X then we can always add a pair (S_j, j) in X with $S_j = \emptyset$.

$$\tilde{\mu} = (\tilde{\mu}(j) = |S_j| \mid j \in Q). \tag{2.26}$$

Since $\chi_{S_j}(i) \leq \hat{z}_P(i,j)$, for all $(i,j) \in E$, the following holds:

(ps′1$_w$) If $S_P \neq \emptyset$ then, for all $i \in S_P$, χ_{S_i} is an optimal solution of (2.4).

By the definitions of p and z_Q, $\nu_{ji}(-p_{ij}) \leq 0$ or $i \in S_j$ for all $(i,j) \in E$ with $z_Q(i,j) = 1$. Thus (ps′2) is satisfied. Also, for any $(i,j) \in E$, if $z_P(i,j) = 0$ then $\nu_{ji}(-p_{ij}) < 0$. In this case, (2.10) implies $p_{ij} = \underline{\pi}_{ij}$ and hence $z_Q(i,j) = 1$ by (2.25). Now, if $z_Q(i,j) = 0$ then (2.25) implies that $p_{ij} = \overline{\pi}_{ij}$. Therefore, $\nu_{ji}(-p_{ij}) \geq 0$ by (2.10). Definition (2.11) yields $z_P(i,j) = 1$. Hence (ps′3) and (ps′4) hold. Therefore, the quadruple $(X; p, z_P, z_Q)$ satisfies (ps′2)$-$(ps′4).

Thus we have found a quadruple $(X; p, z_P, z_Q)$ which satisfies (ps′1$_w$), (ps′2)$-$(ps′4). Our main purpose is to find a quadruple $(X; p, z_P, z_Q)$ which satisfies (ps′1)$-$(ps′4). We observe that if $S_P = P$ then (ps′1$_w$) and (ps′1) coincide. Also, if $\tilde{z}_P(i,j) = 0$ for all $(i,j) \in E$ with $i \in P \setminus S_P$ then again (ps′1$_w$) and (ps′1) coincide. Now, if $\tilde{z}_P(i_0,j) = 1$ for some $(i_0,j) \in E$ with $i_0 \in P \setminus S_P$ then we modify p as well as X, z_P and z_Q in such a way that (ps′1$_w$), (ps′2)$-$(ps′4) are preserved. We express this modification procedure here.

Define a set $E_P \subseteq E$ by

$$E_P = \{(i,j) \in E \mid \hat{z}_P(i,j) = 1\}. \tag{2.27}$$

Construct a directed graph $T = (\{i_0\} \cup E_P, A)$ with arc set A consisting of three disjoint sets of arcs A_0, A_1 and A_2 defined by

$$\begin{aligned} A_0 &:= \{(i_0,(i_0,j)) \mid (i_0,j) \in E_P\}, \\ A_1 &:= \{((i,j),(k,j)) \in E_P \times E_P \mid i \notin S_j, k \in S_j, \\ &\qquad \nu_{jk}(-p_{kj}) = r_j^{S_j} = \nu_{ji}(-p_{ij})\}, \\ A_2 &:= \{((i,j),(i,k)) \in E_P \times E_P \mid i \in S_j, j \neq k\}. \end{aligned} \tag{2.28}$$

For any $((i,j),(i,k)) \in A_2$ note that since $i \in S_j$, therefore, $i \notin S_k$. We assign a weight to each arc of A by defining the weight function $w : A \to \mathbb{R}$ as follows:

$$w(a) = \begin{cases} \ln \alpha_{i_0 j} & \text{if } a = (i_0, (i_0, j)) \in A_0, \\ -\ln \alpha_{ji} + \ln \alpha_{jk} & \text{if } a = ((i,j),(k,j)) \in A_1, \\ -\ln \alpha_{ij} + \ln \alpha_{ik} & \text{if } a = ((i,j),(i,k)) \in A_2. \end{cases} \tag{2.29}$$

We show that the graph T contains no negative cycle with respect to the weight function w. Next lemma is an easy modification of the Lemma 4.2 [15].

Lemma 2.2. *T has no negative cycle with respect to the weight function w.*

Proof. We prove this by contradiction. Suppose that the graph T has a negative cycle C. We express C by a sequence of pairs of E_P as

$$C = (i_1, j_1), (i_2, j_1), (i_2, j_2), \ldots, (i_n, j_n), (i_{n+1}, j_n), (i_{n+1}, j_{n+1}) = (i_1, j_1), \tag{2.30}$$

where $((i_h, j_h), (i_{h+1}, j_h)) \in A_1$ for all $h = 1, 2, \ldots, n$ and $((i_h, j_{h-1}), (i_h, j_h)) \in A_2$ for all $h = 2, 3, \ldots, n+1$. By definition (2.29) of the weight function w, the weight $w(C)$ of the cycle C is calculated as follows:

$$w(C) = \sum_{h=1}^{n} (-\ln \alpha_{j_h i_h} + \ln \alpha_{j_h i_{h+1}} - \ln \alpha_{i_{h+1} j_h} + \ln \alpha_{i_{h+1} j_{h+1}}).$$

Since $i_{n+1} = i_1$ and $j_{n+1} = j_1$, it follows that

$$w(C) = \sum_{h=1}^{n} \left(\ln \alpha_{j_h i_{h+1}} - \ln \alpha_{i_{h+1} j_h} \right) - \sum_{h=1}^{n} \left(\ln \alpha_{j_h i_h} - \ln \alpha_{i_h j_h} \right).$$

By assumption, C is a negative cycle. Therefore, $w(C) < 0$. This implies that

$$\sum_{h=1}^{n} \left(\ln \alpha_{j_h i_{h+1}} - \ln \alpha_{i_{h+1} j_h} \right) - \sum_{h=1}^{n} \left(\ln \alpha_{j_h i_h} - \ln \alpha_{i_h j_h} \right) < 0$$

which further can be written as

$$\sum_{h=1}^{n} \left(\ln \alpha_{j_h i_{h+1}} - \ln \alpha_{i_{h+1} j_h} \right) < \sum_{h=1}^{n} \left(\ln \alpha_{j_h i_h} - \ln \alpha_{i_h j_h} \right). \tag{2.31}$$

By (2.28) and (2.30), it is obvious to see that $(i_h, j_h) \in E_P \cap C$ with $i_h \notin S_{j_h}$ and $i_{h+1} \in S_{j_h}$ for each $h = 1, 2, \ldots, n$. Define

$$\widehat{S}_{j_h} = \{S_{j_h} \setminus \{i_{h+1}\}\} \cup \{i_h\} \quad (\forall h = 1, 2, \ldots, n).$$

Moreover, define $\widetilde{S}_j$ as follows:

$$\widetilde{S}_j = \begin{cases} \widehat{S}_j & \text{if } (i, j) \in C \text{ for some } j \in Q \\ S_j & \text{otherwise} \end{cases} \qquad (\forall j \in Q).$$

We have

$$\sum_{(i,j)\in E} (\ln \alpha_{ji} - \ln \alpha_{ij})\chi_{S_j}(i) = \sum_{(i,j)\in E\cap C} (\ln \alpha_{ji} - \ln \alpha_{ij})\chi_{S_j}(i) + \sum_{(i,j)\in E\setminus C} (\ln \alpha_{ji} - \ln \alpha_{ij})\chi_{S_j}(i).$$

By (2.28) and (2.30), we know that $i_{h+1} \in S_{j_h}$ for each $h = 1, 2, \ldots, n$. Therefore, we may write the above equation as

$$\sum_{(i,j)\in E} (\ln \alpha_{ji} - \ln \alpha_{ij})\chi_{S_j}(i) = \sum_{h=1}^{n} \left(\ln \alpha_{j_h i_{h+1}} - \ln \alpha_{i_{h+1} j_h}\right) + \sum_{(i,j)\in E\setminus C} (\ln \alpha_{ji} - \ln \alpha_{ij})\chi_{S_j}(i).$$

Using the inequality (2.31), we obtain

$$\sum_{(i,j)\in E} (\ln \alpha_{ji} - \ln \alpha_{ij})\chi_{S_j}(i) < \sum_{h=1}^{n} (\ln \alpha_{j_h i_h} - \ln \alpha_{i_h j_h}) + \sum_{(i,j)\in E\setminus C} (\ln \alpha_{ji} - \ln \alpha_{ij})\chi_{S_j}(i).$$

We have $i_h \in \widehat{S}_{j_h}$ for each $h = 1, 2, \ldots, n$, from the above inequality we obtain

$$\sum_{(i,j)\in E} (\ln \alpha_{ji} - \ln \alpha_{ij})\chi_{S_j}(i) < \sum_{(i,j)\in E} (\ln \alpha_{ji} - \ln \alpha_{ij})\chi_{\widetilde{S}_j}(i). \tag{2.32}$$

Since S_j satisfies (2.21) for each $j \in Q$ it follows that $\widetilde{S}_{j_h}$ also satisfies (2.21) for $h = 1, 2, \ldots, n$. This means that the matching $\widetilde{X} = \{(\widetilde{S}_j, j) \mid j \in Q\}$ in

$(\eta, Q; \Gamma)$ satisfies (2.21). By the constructions of graph T, it is also obvious that $\nu_{j_h i_{h+1}}(-p_{i_{h+1} j_h}) = \nu_{j_h i_h}(-p_{i_h j_h})$ for all $h = 1, 2, \ldots, n$. Therefore, $\widetilde{X}$ also satisfies (2.22). Thus the inequality (2.32) contradicts (2.23). Hence, T has no negative cycle. □

By Lemma 2.2, we can find shortest distances from i_0 to all vertices of T. Define $d : E \to \mathbb{R} \cup \{+\infty\}$ where $d(i, j)$ denotes the shortest distance from i_0 to (i, j) with respect to the weight function w in the graph T if $(i, j) \in E_P$, otherwise take $d(i, j) = +\infty$. For any parameter $\varepsilon \geq 0$, we define $p(\varepsilon)$ as follows:

$$p_{ij}(\varepsilon) = p_{ij} - \varepsilon \exp(-d(i, j)) \quad (\forall (i, j) \in E). \tag{2.33}$$

The definition (2.33) states that for any $(i, j) \in E$, p_{ij} is decreased by a parameter $\varepsilon > 0$ if there is a path from i_0 to (i, j) in graph T, that is, $d(i, j) < +\infty$ and remains unchanged otherwise. We denote the set of all reachable vertices from i_0 in the graph T by $R(i_0)$, that is,

$$R(i_0) = \{(i, j) \in E \mid d(i, j) < +\infty\}. \tag{2.34}$$

Observe that $R(i_0) \subseteq E_P$. The following lemmas give few characteristics of the graph T. These lemmas are modifications of the Lemmas 4.1, 4.3 and 4.4 [15] related to our model.

Lemma 2.3. *Assume that $(i, j) \in R(i_0)$. If $i \notin S_j$ then $0 \leq \nu_{ji}(-p_{ij}) \leq r_j$.*

Proof. On contrary, suppose that $\nu_{ji}(-p_{ij}) > r_j$. We assume that j is saturated and $r_j = \nu_{jk}(-p_{kj})$ for some $k \in S_j$. Since (i, j) is reachable from i_0, there exists a sequence S of pairs in E_P given by:

$$S = (i_0, j_0), (i_1, j_0), (i_1, j_1), \ldots, (i_n, j_n) = (i, j), (i_{n+1}, j_n) = (k, j) \tag{2.35}$$

such that $((i_h, j_h), (i_{h+1}, j_h)) \in A_1$ for $h = 0, 1, \ldots, n$ and $((i_h, j_{h-1}), (i_h, j_h)) \in A_2$ for $h = 1, \ldots, n$. By (2.28) and (2.35), obviously $i_{h+1} \in S_{j_h}$ for each $h =$

$0, 1, \ldots, n$. Define

$$\widehat{S}_{j_h} = \{S_{j_h} \setminus \{i_{h+1}\}\} \cup \{i_h\} \quad (\forall h = 0, 1, \ldots, n). \tag{2.36}$$

From (2.36), we see that $i_h \in \widehat{S}_{j_h}$ for each $h = 0, 1, \ldots, n$. By the construction of graph T, we have $\nu_{j_h i_h}(-p_{i_h j_h}) = \nu_{j_h i_{h+1}}(-p_{i_{h+1} j_h})$ for $h = 0, 1, \ldots, n-1$ and $\nu_{j_n i_{n+1}}(-p_{j_n i_{n+1}}) = \nu_{jk}(-p_{kj}) < \nu_{ji}(-p_{ij}) = \nu_{j_n i_n}(-p_{i_n j_n})$. This gives

$$\sum_{h=0}^{n} \nu_{j_h i_{h+1}}(-p_{i_{h+1} j_h}) < \sum_{h=0}^{n} \nu_{j_h i_h}(-p_{i_h j_h})$$

which implies that

$$\sum_{(i,j)\in S} \nu_{ji}(-p_{ij})\chi_{S_j}(i) < \sum_{(i,j)\in S} \nu_{ji}(-p_{ij})\chi_{\widehat{S}_j}(i). \tag{2.37}$$

Define

$$\widetilde{S}_j = \begin{cases} \widehat{S}_j & \text{if } (i,j) \in S \text{ for some } j \in Q \\ S_j & \text{otherwise} \end{cases} \qquad (\forall j \in Q).$$

By definition, $\widetilde{S}_j$ satisfies (2.21) for each $j \in Q$. By (2.37), the matching $\widetilde{X} = \{(\widetilde{S}_j, j) \mid j \in Q\}$ satisfies

$$\sum_{(i,j)\in E} \nu_{ji}(-p_{ij})\chi_{S_j}(i) < \sum_{(i,j)\in E} \nu_{ji}(-p_{ij})\chi_{\widetilde{S}_j}(i).$$

This, however, is a contradiction to (2.22). Thus, the assertion holds in this case.

Now assume that $\nu_{ji}(-p_{ij}) > r_j$ and j is unsaturated. This means that $r_j = 0$ and we have $\nu_{ji}(-p_{ij}) > 0$. Since $(i,j) \in R(i_0)$, there exists a sequence S' of pairs in E_P given by:

$$S' = (i_0, j_0), (i_1, j_0), (i_1, j_1), \ldots, (i_n, j_{n-1}), (i_n, j_n) = (i, j) \tag{2.38}$$

such that $((i_h, j_h), (i_{h+1}, j_h)) \in A_1$ for $h = 0, 1, \ldots, n-1$ and $((i_h, j_{h-1}), (i_h, j_h)) \in A_2$ for $h = 1, \ldots, n$. By (2.28) and (2.38), obviously $i_{h+1} \in S_{j_h}$, for each $h = 0, 1, \ldots, n-1$. Define

$$\widehat{S}'_{j_h} = \{S_{j_h} \setminus \{i_{h+1}\}\} \cup \{i_h\} \quad (\forall h = 0, 1, \ldots, n-1), \tag{2.39}$$

$$\widehat{S}'_{j_n} = S_{j_n} \cup \{i_n\}. \tag{2.40}$$

Moreover, define

$$\widetilde{S}'_j = \begin{cases} \widehat{S}'_j & \text{if } (i,j) \in S \text{ for some } j \in Q \\ S_j & \text{otherwise} \end{cases} \qquad (\forall j \in Q).$$

From (2.39) and (2.40), we see that $i_h \in \widehat{S}'_{j_h}$ for each $h = 0, 1, \ldots, n$. By the construction of graph T, we have $\nu_{j_h i_h}(-p_{i_h j_h}) = \nu_{j_h i_{h+1}}(-p_{i_{h+1} j_h})$ for $h = 0, 1, \ldots, n-1$. Thus

$$\sum_{h=0}^{n-1} \nu_{j_h i_{h+1}}(-p_{i_{h+1} j_h}) = \sum_{h=0}^{n-1} \nu_{j_h i_h}(-p_{i_h j_h}).$$

Since $\nu_{j_n i_n}(-p_{i_n j_n}) = \nu_{ji}(-p_{ij}) > 0$ it follows that

$$\sum_{h=0}^{n-1} \nu_{j_h i_{h+1}}(-p_{i_{h+1} j_h}) < \sum_{h=0}^{n} \nu_{j_h i_h}(-p_{i_h j_h}).$$

We know that $i_n \in \widehat{S}'_{j_n}$ by (2.40), therefore, we have

$$\sum_{(i,j)\in S'} \nu_{ji}(-p_{ij})\chi_{S_j}(i) < \sum_{(i,j)\in S'} \nu_{ji}(-p_{ij})\chi_{\widehat{S}'_j}(i). \tag{2.41}$$

By definition, $\widetilde{S}'_j$ satisfies (2.21) for each $j \in Q$. By (2.41), the matching $\widetilde{X}' = \{(\widetilde{S}'_j, j) \mid j \in Q\}$ satisfies

$$\sum_{(i,j)\in E} \nu_{ji}(-p_{ij})\chi_{S_j}(i) < \sum_{(i,j)\in E} \nu_{ji}(-p_{ij})\chi_{\widetilde{S}'_j}(i).$$

This, however, is a contradiction to (2.22). Thus, the assertion holds. □

Lemma 2.4. *If $(i,j) \in R(i_0)$ then $(i,k) \in R(i_0)$ for all $(i,k) \in E_P$. Furthermore, $\nu_{ij}(p_{ij}(\varepsilon)) = \nu_{ik}(p_{ik}(\varepsilon))$ for any $\varepsilon \geq 0$.*

Proof. If $(i,j) \in R(i_0)$ is such that $i \in S_j$ and $j \neq k$ then $((i,j),(i,k)) \in A_2$, and hence $(i,k) \in R(i_0)$. We assume that $(i,j) \in R(i_0)$ with $i \notin S_j$. If $i = i_0$ then $(i,(i,k)) \in A_0$ for each $(i,k) \in E_P$, that is, $(i,k) \in R(i_0)$ for all $(i,k) \in E_P$. Assume that $i \neq i_0$. Then by the construction of the graph T, there is a unique

$(i, j') \in E_P$ with $i \in S_{j'}$ such that $((i, j'), (i, j)) \in A_2$. Since $(i, j) \in R(i_0)$, therefore, $(i, j') \in R(i_0)$. Hence, $(i, k) \in R(i_0)$ for each $(i, k) \in E_P$.

We next show that $\nu_{ij}(p_{ij}(\varepsilon)) = \nu_{ik}(p_{ik}(\varepsilon))$ for any $\varepsilon \geq 0$. If $i = i_0$ then the unique path from i_0 to (i_0, j) is $(i_0, (i_0, j))$ and hence by (2.33), for all $(i_0, k) \in E_P$, we have

$$\nu_{i_0k}(p_{i_0k}(\varepsilon)) = \nu_{i_0k}(p_{i_0k} - \varepsilon \exp(-d(i_0, k))).$$

On using the definition of linear valuations given by (2.9), we obtain

$$\nu_{i_0k}(p_{i_0k}(\varepsilon)) = \alpha_{i_0k}((p_{i_0k} - \varepsilon \exp(-d(i_0, k))) + \beta_{i_0k}.$$

Since $(i_0, (i_0, k)) \in A_0$, therefore, by (2.29), the weight of the path from i_0 to (i_0, k) is

$$d(i_0, k) = \ln \alpha_{i_0k}.$$

Using this value of $d(i_0, k)$, we obtain

$$\begin{aligned} \nu_{i_0k}(p_{i_0k}(\varepsilon)) &= \alpha_{i_0k}((p_{i_0k} - \varepsilon \exp(-\ln \alpha_{i_0k}) + \beta_{i_0k} \\ &= \nu_{i_0k}(p_{i_0k}) - \varepsilon \\ &= \nu_{i_0j}(p_{i_0j}) - \varepsilon = \nu_{i_0j}(p_{i_0j}(\varepsilon)). \end{aligned}$$

Next we consider the case when $i \neq i_0$. Without loss of generality, assume that $i \in S_j$. For each $(i, k) \in E_P$ with $i \notin S_k$, we have $((i, j), (i, k)) \in A_2$. The shortest distance of (i, k) from i_0 is given by $d(i, k) = d(i, j) + w((i, j), (i, k))$. By using the weight function (2.29), we have

$$d(i, k) = d(i, j) + (-\ln \alpha_{ij} + \ln \alpha_{ik}). \tag{2.42}$$

By (2.33) and (2.42), $\nu_{ik}(p_{ik}(\varepsilon))$ can be written as follows:

$$\begin{aligned} \nu_{ik}(p_{ik}(\varepsilon)) &= \nu_{ik}(p_{ik} - \varepsilon \exp(-d(i, k))) \\ &= \alpha_{ik}(p_{ik} - \varepsilon \exp(-d(i, j) + \ln \alpha_{ij} - \ln \alpha_{ik})) + \beta_{ik} \\ &= \nu_{ik}(p_{ik}) - \varepsilon \alpha_{ij} \exp(-d(i, j)) \\ &= \nu_{ij}(p_{ij}) - \varepsilon \alpha_{ij} \exp(-d(i, j)). \end{aligned}$$

This means that $\nu_{ik}(p_{ik}(\varepsilon)) = \nu_{ij}(p_{ij}(\varepsilon))$ for each $(i,k) \in E_P$. □

Lemma 2.5. *Assume that $(i,j) \in R(i_0)$ with $i \notin S_j$ and there exists $k \in S_j$ such that $r_j^{S_j} = \nu_{jk}(-p_{kj})$. Then the following statements hold:*

(i) $\nu_{ji}(-p_{ij}(\varepsilon)) \leq \nu_{jk}(-p_{kj}(\varepsilon))$ *for a sufficiently small $\varepsilon \geq 0$.*

(ii) $\nu_{ji}(-p_{ij}(\varepsilon)) \leq \nu_{jk}(-p_{kj}(\varepsilon))$ *for all $\varepsilon \geq 0$ if $((i,j),(k,j)) \in A_1$.*

(iii) $\nu_{ji}(-p_{ij}(\varepsilon)) = \nu_{jk}(-p_{kj}(\varepsilon))$ *for all $\varepsilon \geq 0$ if the arc $((i,j),(k,j))$ lies on a shortest path from i_0 to (k,j).*

Proof. **(i)** By Lemma 2.3, we have $\nu_{ji}(-p_{ij}) \leq r_j$. By (2.2) and (2.15), we have $r_j \leq r_j^{S_j}$. Thus $\nu_{ji}(-p_{ij}) \leq r_j^{S_j} = \nu_{jk}(-p_{kj})$. If $\nu_{ji}(-p_{ij}) < \nu_{jk}(-p_{kj})$, then assertion obviously holds. Otherwise, $\nu_{ji}(-p_{ij}) = \nu_{jk}(-p_{kj})$, that is, $((i,j),(k,j)) \in A_1$. We consider this case in (ii).

(ii) Let $((i,j),(k,j)) \in A_1$. Since d is the shortest distance with respect to w, we have $d(k,j) \leq d(i,j) - \ln \alpha_{ji} + \ln \alpha_{jk}$. Hence, we have

$$\begin{aligned}
\nu_{jk}(-p_{kj}(\varepsilon)) &= \alpha_{jk}(-p_{kj} + \varepsilon \exp(-d(k,j))) + \beta_{jk} \\
&= \nu_{ji}(-p_{ij}) + \varepsilon \alpha_{jk} \exp(-d(k,j)) \\
&\geq \nu_{ji}(-p_{ij}) + \varepsilon \alpha_{jk} \exp(-d(i,j) + \ln \alpha_{ji} - \ln \alpha_{jk}) \;=\; \nu_{ji}(-p_{ij}(\varepsilon)).
\end{aligned}$$

(iii) Suppose that $((i,j),(k,j)) \in A_1$ lies on a shortest path from i_0 to (k,j), then $d(k,j) = d(i,j) - \ln \alpha_{ji} + \ln \alpha_{jk}$. Hence, we have

$$\begin{aligned}
\nu_{jk}(-p_{kj}(\varepsilon)) &= \alpha_{jk}(-p_{kj} + \varepsilon \exp(-d(k,j))) + \beta_{jk} \\
&= \nu_{ji}(-p_{ij}) + \varepsilon \alpha_{jk} \exp(-d(k,j)) \\
&= \nu_{ji}(-p_{ij}) + \varepsilon \alpha_{jk} \exp(-d(i,j) + \ln \alpha_{ji} - \ln \alpha_{jk}) \\
&= \nu_{ji}(-p_{ij}(\varepsilon)).
\end{aligned}$$

□

Our aim is to propose an algorithm which finds a stable job allocation. In each iteration of the algorithm, we shall modify p by a parameter ε in such a way that the conditions (ps$'1_w$), (ps$'2$)$-$(ps$'4$) are preserved. All possible cases which may arise by modifying p are discussed below:

Case 1. For any $(i,j) \in R(i_0)$, we have $\nu_{ij}(p_{ij}) \geq \nu_{ik}(p_{ik})$ for all $(i,k) \in E$. If $(i,j),(i,k) \in R(i_0)$ then $\nu_{ij}(p_{ij}) = \nu_{ik}(p_{ik})$. Lemma 2.4 implies that $\nu_{ij}(p_{ij}(\varepsilon)) = \nu_{ik}(p_{ik}(\varepsilon))$ for all $\varepsilon \geq 0$. If $(i,j) \in R(i_0)$ and $(i,k) \notin R(i_0)$ then $\nu_{ij}(p_{ij}) > \nu_{ik}(p_{ik})$. Hence, we can find $\varepsilon > 0$ such that $\nu_{ij}(p_{ij}(\varepsilon)) = \nu_{ik}(p_{ik}(\varepsilon)) = \nu_{ik}(p_{ik})$. In this way, a new element (i,k) is added in E_P which may augment S_k. This, however, depends upon whether $|S_k| = \mu(k)$ or $|S_k| < \mu(k)$.

Case 2. Since valuations are strictly increasing, for any $(i,j) \in R(i_0)$ we can always find an $\varepsilon \geq 0$ such that $\nu_{ij}(p_{ij}(\varepsilon)) = 0$ or $p_{ij}(\varepsilon) = \underline{\pi}_{ij} > -\infty$. As mentioned earlier, we want (ps$'1$) and (ps$'1_w$) to coincide. If $\tilde{z}_P$ is decreased, (ps$'1$) and (ps$'1_w$) comes closer to each other. Now if $\nu_{ij}(p_{ij}(\varepsilon)) = 0$ then we can decrease z_0. Consequently, $\tilde{z}_P$ will decrease. If $p_{ij}(\varepsilon) = \underline{\pi}_{ij}$ then decreasing $p(\varepsilon)$ further would mean that it is no longer a feasible salary vector. Also, in this case, we can switch $z_P(i,j)$ to zero and $z_Q(i,j)$ to 1. Thus (ps$'3$) and (ps$'4$) are preserved, and $\tilde{z}_P$ will decrease.

Case 3. Let $(i,j) \in R(i_0)$ and $((i,j),(k,j)) \in A_1$. Without loss of generality, assume that $((i,j),(k,j))$ lies on the shortest path from i_0 to (k,j). Suppose that there exists $k' \in S_j \setminus \{k\}$. Then $r_j^{S_j} \leq \nu_{jk'}(-p_{k'j})$.

(i) If $r_j^{S_j} = \nu_{jk'}(-p_{k'j})$ then $((i,j),(k',j)) \in A_1$. By the construction of graph T, $((i,j),(k',j))$ lies on the shortest path from i_0 to (k',j). Therefore, for any $\varepsilon \geq 0$, $\nu_{jk}(-p_{kj}(\varepsilon)) = \nu_{jk'}(-p_{k'j}(\varepsilon))$ by Lemma 2.5.

(ii) If $r_j^{S_j} < \nu_{jk'}(-p_{k'j})$ then again by the construction of the graph T, $(k',j) \notin R(i_0)$. Hence, one can find $\varepsilon > 0$ such that $\nu_{jk}(-p_{kj}(\varepsilon)) = \nu_{jk'}(-p_{k'j}(\varepsilon))$. By Lemma 2.5, we get $\nu_{ji}(-p_{ij}(\varepsilon)) = \nu_{jk}(-p_{kj}(\varepsilon))$. Thus any $\varepsilon' > \varepsilon$ yields $\nu_{ji}(-p_{ij}(\varepsilon')) = \nu_{jk}(-p_{kj}(\varepsilon')) > \nu_{jk'}(-p_{k'j})$. But $\nu_{jk'}(-p_{k'j}) = \nu_{jk'}(-p_{k'j}(\varepsilon'))$ since $(k',j) \notin R(i_0)$. This shows that (ps$'2$) does not hold. Therefore, $p(\varepsilon)$ cannot be decreased any more.

Case 4. Let $(i,j) \in R(i_0)$ with $i \notin S_j$ and there exists $k \in S_j$ such that $r_j^{S_j} = \nu_{jk}(-p_{kj})$. From (2.2) and (2.15), it is easy to see that $r_j \leq r_j^{S_j}$. Then $\nu_{ji}(-p_{ij}) \leq r_j^{S_j}$ by Lemma 2.3.

(i) If $\nu_{ji}(-p_{ij}) = \nu_{jk}(-p_{kj})$ then $((i,j),(k,j)) \in A_1$. Lemma 2.5 implies that $\nu_{ji}(-p_{ij}(\varepsilon)) \leq \nu_{jk}(-p_{kj}(\varepsilon))$ for any $\varepsilon \geq 0$.

(ii) If $\nu_{ji}(-p_{ij}) < \nu_{jk}(-p_{kj})$ and $d(i,j) \geq d(k,j)$ then $\nu_{ji}(-p_{ij}(\varepsilon)) < \nu_{jk}(-p_{kj}(\varepsilon))$ for any $\varepsilon \geq 0$.

(iii) If $\nu_{ji}(-p_{ij}) < \nu_{jk}(-p_{kj})$ and $d(i,j) < d(k,j)$ then one can find an $\varepsilon > 0$ such that $\nu_{ji}(-p_{ij}(\varepsilon)) = \nu_{jk}(-p_{kj}(\varepsilon))$. Then $\nu_{ji}(-p_{ij}(\varepsilon')) > \nu_{jk}(-p_{kj}(\varepsilon'))$ for any $\varepsilon' > \varepsilon$. This, however, shows that (ps′2) does not hold. Therefore, in this case we cannot decrease $p(\varepsilon)$ any more.

With the discussion above, we define the following parameters.

$$
\begin{aligned}
\varepsilon_1 &= \max\{\varepsilon \geq 0 \mid \nu_{ij}(p_{ij}(\varepsilon)) \geq \nu_{ik}(p_{ik}) \quad \forall (i,j) \in R(i_0), \\
&\qquad \forall (i,k) \in E \text{ with } z_P(i,k) = 1 \text{ and } |S_k| < \mu(k)\}, \\
\varepsilon_2 &= \max\{\varepsilon \geq 0 \mid \nu_{ij}(p_{ij}(\varepsilon)) \geq \nu_{ik}(p_{ik}) \quad \forall (i,j) \in R(i_0), \\
&\qquad \forall (i,k) \in E \setminus R(i_0) \text{ with } z_P(i,k) = 1 \text{ and } |S_k| = \mu(k)\}, \\
\varepsilon_3 &= \max\{\varepsilon \geq 0 \mid \nu_{ij}(p_{ij}(\varepsilon)) \geq 0 \quad \forall (i,j) \in R(i_0)\}, \\
\varepsilon_4 &= \max\{\varepsilon \geq 0 \mid p_{ij}(\varepsilon) \geq \underline{\pi}_{ij} \quad \forall (i,j) \in R(i_0)\}, \\
\varepsilon_5 &= \max\{\varepsilon \geq 0 \mid \nu_{ji}(-p_{ij}(\varepsilon)) \leq \nu_{jk}(-p_{kj}) \quad \forall (i,j) \in R(i_0), \\
&\qquad \forall (k,j) \in E_P \text{ with } i,k \in S_j \text{ and } \nu_{ji}(-p_{ij}) < \nu_{jk}(-p_{kj})\}, \\
\varepsilon_6 &= \max\{\varepsilon \geq 0 \mid \nu_{ji}(-p_{ij}(\varepsilon)) \leq \nu_{jk}(-p_{kj}(\varepsilon)) \quad \forall (i,j) \in R(i_0), \\
&\qquad \forall (k,j) \in E_P \text{ with } i \notin S_j, k \in S_j, d(i,j) < d(k,j) \\
&\qquad \text{ and } \nu_{ji}(-p_{ij}) < \nu_{jk}(-p_{kj})\},
\end{aligned}
\tag{2.43}
$$

where the maximum over an empty set is defined to be $+\infty$. Observe that the existence of ε_1 and ε_2 is due to Case 1, ε_3 and ε_4 is due to Case 2, ε_5 is due to

Case 3 and ε_6 is due to Case 4. To modify the salary vector p we determine the parameter $\varepsilon \geq 0$ by

$$\varepsilon = \min\{\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, \varepsilon_5, \varepsilon_6\}. \tag{2.44}$$

Note that ε is well defined by the definition of ε_3. For the sake of convenience, we assume that ε_1 and ε_2 exist for $(\hat{i}, \hat{j}), (\hat{i}, \hat{k}) \in E$, ε_3 and ε_4 exist for $(\hat{i}, \hat{j}) \in E$ and, ε_5 and ε_6 exist for $(\hat{i}, \hat{j}), (\hat{k}, \hat{j}) \in E$. By the above definition of ε, if $\varepsilon = \varepsilon_1$ or $\varepsilon = \varepsilon_2$ then $\nu_{\hat{i}\hat{j}}(p_{\hat{i}\hat{j}}(\varepsilon)) = \nu_{\hat{i}\hat{k}}(p_{\hat{i}\hat{k}}(\varepsilon))$. Similarly, if $\varepsilon = \varepsilon_3$ then $\nu_{\hat{i}\hat{j}}(p_{\hat{i}\hat{j}}(\varepsilon)) = 0$ and if $\varepsilon = \varepsilon_4$ then $p_{\hat{i}\hat{j}}(\varepsilon) = \underline{\pi}_{\hat{i}\hat{j}}$. Also, if $\varepsilon = \varepsilon_5$ or $\varepsilon = \varepsilon_6$ then $\nu_{\hat{j}\hat{i}}(-p_{\hat{i}\hat{j}}(\varepsilon)) = \nu_{\hat{j}\hat{k}}(-p_{\hat{k}\hat{j}}(\varepsilon))$.

Next we describe the algorithm which finds a stable outcome in a finite number of iterations.

Algorithm `Job_Allocation`

Step 0: Initially, define $p, z_P, z_0, \tilde{z}_P, \hat{z}_P$ by (2.10)$-$(2.14) and $z_Q, \tilde{\mu}, \eta_j (j \in Q), \eta, \Gamma$ by (2.16)$-$(2.20).

Step 1: Find a matching X in the bipartite graph $(\eta, Q; \Gamma)$ satisfying (2.21)$-$(2.23). Define S_P by (2.24) and update z_Q and $\tilde{\mu}$ by (2.25) and (2.26).

Step 2: If $P = S_P$ or for any $i \in P \setminus S_P$ we have $\tilde{z}_P(i, j) = 0$, for all $(i, j) \in E$, then stop.

Step 3: Let $i_0 \in P \setminus S_P$ such that $\tilde{z}_P(i_0, j) = 1$ for some $j \in Q$. Define E_P by (2.27).

Step 4: Construct a directed graph $T = (\{i_0\} \cup E_P, A)$ with arc set A consisting of A_0, A_1 and A_2 defined by (2.28). Define weight function w by (2.29). Find the shortest distances $d(i, j)$ from i_0 to all vertices (i, j) of T with respect to w and put $d(i, j) = +\infty$ if (i, j) is not a vertex of T. Determine ε by (2.44), find $p(\varepsilon)$ by (2.33) and define $R(i_0)$ by (2.34).

Step 5: **(a)** If $\varepsilon = \varepsilon_1$ then set $\tilde{\mu}(\hat{k}) := \tilde{\mu}(\hat{k}) + 1$ and go to Step 7; else go to (b).

(b) If $\varepsilon = \varepsilon_2$ then go to Step 7; else go to (c).

(c) If $\varepsilon = \varepsilon_3$ then set $z_0(\hat{\imath}, j) := 0$ for all $j \in Q$ and go to Step 7; else go to (d).

(d) If $\varepsilon = \varepsilon_4$ then set $z_P(\hat{\imath}, \hat{\jmath}) := 0$ and $z_Q(\hat{\imath}, \hat{\jmath}) := 1$ and go to Step 7; else go to (e).

(e) If $\varepsilon = \varepsilon_5$ then go to Step 7; else go to (f).

(f) If $\varepsilon = \varepsilon_6$ then, for each $j \in Q$, define $r_j^{S_j}$ by (2.15) for $p(\varepsilon)$. Construct a directed graph $T = (\{i_0\} \cup E_P, A)$ with arc set A consisting of A_0, A_1 and A_2 defined by (2.28) for $p(\varepsilon)$. Define weight function w by (2.29) and $R(i_0)$ by (2.34). If X satisfies (2.22) and (2.23) then put $p = p(\varepsilon)$ and go to Step 6; else go to Step 7.

Step 6: Update z_Q by (2.25). Find the shortest distances $d(i, j)$ from i_0 to all vertices (i, j) of T with respect to w and put $d(i, j) = +\infty$ if (i, j) is not a vertex of T. Determine ε by (2.44) and find $p(\varepsilon)$ by (2.33). Go to Step 5.

Step 7: Put $p := p(\varepsilon)$. Update $\tilde{z}_P$ and $\hat{z}_P$ by (2.13)$-$(2.14) and define η_j ($j \in Q$), η and Γ by (2.18)$-$(2.20). Go to Step 1.

To explain `Job_Allocation`, we present an example.

Example 2.6. Let $P = \{i_0, i_1, i_2\}$, $Q = \{j_0, j_1, j_2\}$, and $E = P \times Q$. Define the upper and lower bounds on the salaries by

$$\overline{\pi}_{ij} = 1, \quad \underline{\pi}_{ij} = -2 \quad (\forall (i, j) \in E).$$

We assume that

$$\mu = (1, \ldots, 1) \in \mathbb{Z}_+^Q.$$

Define valuations by (2.9) with α_{ij}, α_{ji}, β_{ij} and β_{ji}, for all $(i, j) \in E$, given below:

$$\alpha_{ij} = \alpha_{ji} = 1 \quad (\forall (i, j) \in E),$$

β_{ij}	j_0	j_1	j_2
i_0	4	2	-2
i_1	4	4	2
i_2	2	4	-2

Table 2.1: β_{ij} for $(i, j) \in E$

β_{ji}	i_0	i_1	i_2
j_0	4	4	4
j_1	10	5	5
j_2	-3	10	-3

Table 2.2: β_{ji} for $(i, j) \in E$.

We begin with Step 0 by defining the salary vector p by (2.10). We find that $\nu_{j_2 i_0}(-\overline{\pi}_{i_0 j_2})$ and $\nu_{j_2 i_2}(-\overline{\pi}_{i_2 j_2})$ are negative and $\nu_{ji}(-\overline{\pi}_{ij}) \geq 0$ for each $(i, j) \in E \setminus \{(i_0, j_2), (i_2, j_2)\}$. By (2.10), we obtain the salary vector p as follows:

$$p_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \setminus \{(i_0, j_2), (i_2, j_2)\} \\ -2 & \text{otherwise} \end{cases} \qquad (\forall (i,j) \in E).$$

Using the values of α_{ij}, α_{ji}, β_{ij}, β_{ji} and p_{ij} we obtain $\nu_{ij}(p_{ij})$ and $\nu_{ji}(-p_{ij})$ for each $(i, j) \in E$ as follows:

$\nu_{ij}(p_{ij})$	j_0	j_1	j_2
i_0	5	3	-4
i_1	5	5	3
i_2	3	5	-4

Table 2.3: $\nu_{ij}(p_{ij})$ for $(i, j) \in E$

$\nu_{ji}(-p_{ij})$	i_0	i_1	i_2
j_0	3	3	3
j_1	9	4	4
j_2	-1	9	-1

Table 2.4: $\nu_{ji}(-p_{ij})$ for $(i, j) \in E$

Also we define the vectors z_P and z_0 by (2.11) and (2.12), respectively. By (2.11), it is obvious that for each $(i, j) \in E$ with $\nu_{ji}(-p_{ij}) \geq 0$, we have $z_p(i, j) = 1$. From the values given in Table 2.4, we find that $\nu_{j_2 i_0}(-p_{i_0 j_2})$ and $\nu_{j_2 i_2}(-p_{i_2 j_2})$ are negative, therefore, we get

$$z_P(i,j) = \begin{cases} 1 & \text{if } (i,j) \in E \setminus \{(i_0, j_2), (i_2, j_2)\} \\ 0 & \text{otherwise} \end{cases} \qquad (\forall (i,j) \in E).$$

By (2.12), we obtain

$$z_0(i,j) = \begin{cases} 1 & \text{if } (i,j) \in E \setminus \{(i_0,j_2),(i_2,j_2)\} \\ 0 & \text{otherwise} \end{cases} \qquad (\forall (i,j) \in E)$$

because from Table 2.3 we see that $\nu_{i_0j_2}(p_{i_0j_2})$ and $\nu_{i_0j_2}(p_{i_0j_2})$ are negative. Then by (2.13), we get $\tilde{z}_P = z_P = z_0$ and, by (2.14), we obtain

$$\hat{z}_P(i,j) = \begin{cases} 1 & \text{if } (i,j) \in \{(i_0,j_0),(i_1,j_0),(i_1,j_1),(i_2,j_1)\} \\ 0 & \text{otherwise} \end{cases} \qquad (\forall (i,j) \in E).$$

We have z_Q by (2.16) as follows:

$$z_Q(i,j) = \begin{cases} 1 & \text{if } (i,j) \in \{(i_0,j_2),(i_2,j_2)\} \\ 0 & \text{otherwise} \end{cases} \qquad (\forall (i,j) \in E).$$

We find $\tilde{\mu} = \mathbf{0}$ by (2.17). Next we calculate η_j for all $j \in Q$ and Γ by (2.18)$-$(2.20) and get

$$\eta_{j_0} = \{\{i_0\},\{i_1\}\}, \quad \eta_{j_1} = \{\{i_1\},\{i_2\}\}, \quad \eta_{j_2} = \{\emptyset\},$$
$$\eta = \{\emptyset,\{i_0\},\{i_1\},\{i_2\}\},$$
$$\Gamma = \{(\{i_0\},j_0),(\{i_1\},j_0),(\{i_1\},j_1),(\{i_2\},j_1),(\emptyset,j_2)\}.$$

Now we go to Step 1. We find a matching X in the bipartite graph $(\eta, Q; \Gamma)$ satisfying (2.21)$-$(2.23). A matching $X = \{(S_{j_0},j_0),(S_{j_1},j_1),(S_{j_2},j_2)\}$ in the bipartite graph $(\eta, Q; \Gamma)$ with $S_{j_0} = \{i_1\}$, $S_{j_1} = \{i_2\}$, $S_{j_2} = \emptyset$ satisfies (2.21)$-$(2.23). By defining S_P by (2.24), z_Q by (2.25) and $\tilde{\mu}$ by (2.26), we obtain

$$S_P = \{i_1, i_2\},$$
$$z_Q(i,j) = \begin{cases} 1 & \text{if } (i,j) \in \{(i_0,j_2),(i_1,j_0),(i_2,j_1),(i_2,j_2)\} \\ 0 & \text{otherwise} \end{cases} \qquad (\forall (i,j) \in E),$$
$$\tilde{\mu} = (1,1,0).$$

This completes Step 1 of `Job_Allocation`. It is now easy to see that the quadruple $(X; p, z_P, z_Q)$ satisfies (ps$'1_w$), (ps$'2$)$-$(ps$'4$). The quadruple $(X; p, z_P, z_Q)$ does not satisfy (ps′1), since $i_0 \in P \setminus S_P$ and $\tilde{z}_P(i_0, j_0) = 1$, it follows that $\chi_{S_{i_0}}$ is not an optimal solution for (ps′1). The `Job_Allocation` does not Stop at Step 2. At Step 3, we calculate E_P by (2.27) which is given by

$$E_P = \{(i_0, j_0), (i_1, j_0), (i_1, j_1), (i_2, j_1)\}.$$

We find a parameter $\varepsilon \geq 0$ to modify salary vector p by the aforementioned procedure. The modification of the salary vector p is done at Step 4. Here we define A_0, A_1 and A_2 by (2.28) and get:

$$\begin{aligned} A_0 &= \{(i_0, (i_0, j))\}, \\ A_1 &= \{((i_0, j_0), (i_1, j_0)), ((i_1, j_1), (i_2, j_1))\}, \\ A_2 &= \{((i_1, j_0), (i_1, j_1))\}. \end{aligned}$$

Construct the graph $T = (\{i_0\} \cup E_P, A)$, where A consists of A_0, A_1 and A_2. Note that weight of each arc of A is zero if we define the weight function w by (2.29). For each arc of T there is a unique path from i_0. Let $d(i, j)$ denote the shortest distance with respect to w from i_0 to (i, j) if $(i, j) \in E_P$ otherwise we set $d(i, j) = +\infty$. Thus we get

$$d(i,j) = \begin{cases} 0 & \text{if } (i,j) \in \{(i_0, j_0), (i_1, j_0), (i_1, j_1), (i_2, j_1)\} \\ +\infty & \text{otherwise} \end{cases} \qquad (\forall (i,j) \in E).$$

Calculating parameters ε_1 to ε_6 by (2.43), we get

$$\varepsilon_1 = \varepsilon_2 = 2, \varepsilon_3 = 5, \varepsilon_4 = 3, \varepsilon_5 = \varepsilon_6 = +\infty.$$

Using (2.44) we have $\varepsilon = 2$. By defining $p(\varepsilon)$ by (2.33), we get

$$p_{ij}(\varepsilon) := \begin{cases} -1 & \text{if } (i,j) \in \{(i_0, j_0), (i_1, j_0), \\ & \quad (i_1, j_1), (i_2, j_1)\} \\ p_{ij} & \text{otherwise} \end{cases} \qquad (\forall (i,j) \in E). \tag{2.45}$$

$\nu_{ij}(p_{ij}(\varepsilon))$	j_0	j_1	j_2
i_0	3	3	-4
i_1	3	3	3
i_2	3	3	-4

Table 2.5: $\nu_{ij}(p_{ij}(\varepsilon))$ for $(i,j) \in E$

$\nu_{ji}(-p_{ij}(\varepsilon))$	i_0	i_1	i_2
j_0	5	5	3
j_1	9	6	6
j_2	-1	9	-1

Table 2.6: $\nu_{ji}(-p_{ij}(\varepsilon))$ for $(i,j) \in E$

For this modified salary vector $\nu_{ij}(p_{ij}(\varepsilon))$ and $\nu_{ji}(-p_{ij}(\varepsilon))$ for each $(i,j) \in E$ are given as follows:

We define $R(i_0)$ by (2.3), we get

$$R(i_0) = \{(i_0, j_0), (i_1, j_0), (i_1, j_1), (i_2, j_1)\}.$$

At Step 5, we see that $\varepsilon = \varepsilon_1 = \varepsilon_2$. Now we go to Step 7. Setting $p := p(\varepsilon)$ and using the values in Table 2.5 and Table 2.6 we update vectors z_P, z_0, $\tilde{z}_P$ and $\hat{z}_P$ for the modified salary vector p. We observe that z_P, z_0 and $\tilde{z}_P$ remain unchanged. However, $\hat{z}_P$ changes. The updated $\hat{z}_P$ by (2.14) is given by:

$$\hat{z}_P(i,j) = \begin{cases} 1 & \text{if } (i,j) \in E \setminus \{(i_0, j_2), (i_2, j_2)\} \\ 0 & \text{otherwise} \end{cases} \qquad (\forall (i,j) \in E).$$

We update η_j, for all $j \in Q$, η and Γ by (2.18)$-$(2.20) for $p(\varepsilon)$ and get

$$\eta_{j_0} = \{\{i_0\}, \{i_1\}\}, \eta_{j_1} = \{\{i_0\}, \{i_1\}, \{i_2\}\}, \eta_{j_2} = \{\{i_1\}\},$$

$$\eta = \{\emptyset, \{i_0\}, \{i_1\}, \{i_2\}\},$$

$$\Gamma = \{(\{i_0\}, j_0), (\{i_1\}, j_0), (\{i_0\}, j_1), (\{i_1\}, j_1), (\{i_2\}, j_1), (\{i_1\}, j_2)\}.$$

Thus we get a unique matching $X = \{(S_{j_0}, j_0), (S_{j_1}, j_1), (S_{j_2}, j_2)\}$ in the bipartite graph $(\eta, Q; \Gamma)$ with $S_{j_0} = \{i_0\}$, $S_{j_1} = \{i_2\}$, $S_{j_2} = \{i_1\}$ which satisfies (2.21)−(2.23). Updating S_P by (2.24), z_Q by (2.25) and $\tilde{\mu}$ by (2.26), we get

$$\begin{aligned} S_P &= P, \\ z_Q(i,j) &= \begin{cases} 1 & \text{if } (i,j) \in E \setminus \{(i_0, j_1), (i_2, j_0)\} \\ 0 & \text{otherwise} \end{cases} \qquad (\forall (i,j) \in E), \\ \tilde{\mu} &= (1,1,1). \end{aligned}$$

Since $S_P = P$, the `Job_Allocation` will stop at Step 2 in the second iteration. One can easily see that the outcome $(X; p(\varepsilon), z_P, z_Q)$ satisfies (ps′1)−(ps′4). By Theorem 2.1, X is a stable job allocation. □

2.6.1 Analysis of the algorithm

Here we show some important properties of the `Job_Allocation`. These properties are then utilized to prove correctness and termination of the `Job_Allocation`.

In the sequel, we shall use the notation [Step AA → Step BB] which means the `Job_Allocation` goes from Step AA to Step BB.

The following lemma describes the important features of `Job_Allocation`.

Lemma 2.7. *In each iteration of* `Job_Allocation`*, the following statements hold:*

(i) *z_P and z_0 decrease or remain the same. z_Q and $\tilde{\mu}$ increase or remain the same.*

(ii) *If $\varepsilon \in \{\varepsilon_1, \varepsilon_2\}$ or $\varepsilon = \varepsilon_5 < \min\{\varepsilon_3, \varepsilon_4\}$ or [Step 5 (f) → Step 7] is executed then $\hat{z}_P$ increases or remains the same. In particular, $\hat{z}_P$ increases if $\varepsilon = \varepsilon_1 > 0$ or $\varepsilon = \varepsilon_2 < \varepsilon_1$.*

(iii) *If $\varepsilon = \varepsilon_3 < \min\{\varepsilon_1, \varepsilon_2\}$ or $\varepsilon = \varepsilon_4 < \min\{\varepsilon_1, \varepsilon_2, \varepsilon_3\}$ then $\hat{z}_P$ decreases.*

(iv) *If [Step 5 (f) $\rightarrow$ Step 6] is executed then the graph* $(\eta, Q; \Gamma)$ *is preserved.*

(v) *If* $\hat{z}_P(i,j)$ *turns to* 0 *from* 1 *at Step 7, for some* $(i,j) \in E$*, then it never changes its orientation in the subsequent iterations.*

(vi) *If* $\hat{z}_P(i,j)$ *turns to* 1 *from* 0 *at Step 7, for some* $(i,j) \in E$*, then* $p_{ij}(\varepsilon)$ *is the initial value defined in* (2.10)*.*

(vii) *For any* $(i,j) \in E$*, if* $p_{ij}(\varepsilon) < p_{ij}$ *at Step 4 or at Step 6 then* $\tilde{\mu}(j) = \mu(j)$*.*

Proof. The following inequality holds for any $(i,j) \in R(i_0)$:

$$\nu_{ij}(p_{ij}(\varepsilon)) \geq \nu_{ik}(p_{ik}(\varepsilon)) \qquad (\forall (i,k) \in E \setminus R(i_0) \text{ with } z_P(i,k) = 1) \tag{2.46}$$

since $\varepsilon \leq \min\{\varepsilon_1, \varepsilon_2\}$. If $\varepsilon < \min\{\varepsilon_1, \varepsilon_2\}$ then

$$\nu_{ij}(p_{ij}(\varepsilon)) > \nu_{ik}(p_{ik}(\varepsilon)) \qquad (\forall (i,k) \in E \setminus R(i_0) \text{ with } z_P(i,k) = 1). \tag{2.47}$$

Also, for all $(i,j), (i,k) \in R(i_0)$, Lemma 2.4 yields that

$$\nu_{ij}(p_{ij}(\varepsilon)) = \nu_{ik}(p_{ik}(\varepsilon)). \tag{2.48}$$

For all $(i,j) \in R(i_0)$, (2.46) and (2.48) imply that

$$\nu_{ij}(p_{ij}(\varepsilon)) \geq \nu_{ik}(p_{ik}(\varepsilon)) \qquad (\forall (i,k) \in E \text{ with } z_P(i,k) = 1). \tag{2.49}$$

(i) z_0 decreases at Step 5 (c) if [Step 5 (c) $\rightarrow$ Step 7] is executed, else it is not updated. Similarly, z_P decreases at Step 5 (d) if [Step 5 (d) $\rightarrow$ Step 7] is executed, else it is not updated. This implies the first part of the assertion.

In each iteration at Step 1, $\tilde{\mu}$ is updated by (2.26) for the current matching where for each $j \in Q$, $\tilde{\mu}(j)$ is the number of workers employed by j. $\tilde{\mu}(\hat{k})$ increases at Step 5 (a) if [Step 5 (a) $\rightarrow$ Step 7] is executed. In the next iteration, $\tilde{\mu}(j)$ is the lower bound of the number of workers employed by j, for each $j \in Q$. Hence $\tilde{\mu}$

increases or remains the same. Next, we see that in each iteration, z_Q is updated at Step 1 and in some iterations it is updated at Step 6 as well. In either case, we use (2.25) to update z_Q. From (2.25), it is obvious to see that z_Q increases or remains the same.

(ii) Observe that $\varepsilon = \varepsilon_1$ if [Step 5 (a) $\rightarrow$ Step 7] is executed. If $\varepsilon = \varepsilon_1 = 0$ then $p_{ij}(\varepsilon) = p_{ij}$, for all $(i, j) \in E$, and hence $\hat{z}_P$ remains the same. If $\varepsilon = \varepsilon_1 > 0$ then $\nu_{\hat{i}\hat{j}}(p_{\hat{i}\hat{j}}) > \nu_{\hat{i}\hat{k}}(p_{\hat{i}\hat{k}})$ and $\nu_{\hat{i}\hat{j}}(p_{\hat{i}\hat{j}}(\varepsilon)) = \nu_{\hat{i}\hat{k}}(p_{\hat{i}\hat{k}}(\varepsilon))$. This together with the inequality (2.49) implies that $\hat{z}_P$ increases. Analogously, we can prove that $\hat{z}_P$ increases when [Step 5 (b) $\rightarrow$ Step 7] is executed. Note that in this case $0 < \varepsilon = \varepsilon_2 < \varepsilon_1$. Next, when [Step 5 (e) $\rightarrow$ Step 7] or [Step 5 (f) $\rightarrow$ Step 7] is executed then $\varepsilon < \min\{\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4\}$. From (2.47) and (2.48), it is clear that $\hat{z}_P$ remains the same.

(iii) [Step 5 (c) $\rightarrow$ Step 7] is executed when $\varepsilon = \varepsilon_3 < \min\{\varepsilon_1, \varepsilon_2\}$ and [Step 5 (d) $\rightarrow$ Step 7] is executed when $\varepsilon = \varepsilon_4 < \min\{\varepsilon_1, \varepsilon_2, \varepsilon_3\}$. Note that by (i), z_0 and z_P remain the same or decrease. If [Step 5 (c) $\rightarrow$ Step 7] is executed then z_0 decreases at Step 5 (c) and if [Step 5 (d) $\rightarrow$ Step 7] is executed then z_P decreases at Step 5 (d). In either case, (2.47) and (2.49) imply that $\hat{z}_P$ decreases at Step 7.

(iv) If [Step 5 (f) $\rightarrow$ Step 6] is executed then $\varepsilon = \varepsilon_6 < \min\{\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, \varepsilon_5\}$. Therefore at Step 6, $\tilde{\mu}$ remains unchanged. Also by (ii), $\hat{z}_P$ remains the same or increases. But from the definitions of ε_1 to ε_6, we observe that $\hat{z}_P$ increases if and only if $\varepsilon = \varepsilon_1 > 0$ or $\varepsilon = \varepsilon_2 < \varepsilon_1$. Therefore $\hat{z}_P$ also remains unchanged when [Step 5 (f) $\rightarrow$ Step 6] is executed. Since $\varepsilon = \varepsilon_6 < \varepsilon_5$, the definitions of ε_5 and ε_6 together with Lemma 2.5 imply the (a) of (BC). The (b) of (BC) holds since $\varepsilon < \varepsilon_1$. Thus $(\eta, Q; \Gamma)$ is preserved at Step 6.

(v) If $\varepsilon = \varepsilon_3 < \min\{\varepsilon_1, \varepsilon_2\}$ or $\varepsilon = \varepsilon_4 < \min\{\varepsilon_1, \varepsilon_2, \varepsilon_3\}$ then $\hat{z}_P$ decreases. In other cases, it increases or remains the same. $\hat{z}_P$ decreases if z_0 decreases at Step 5 (c) or z_P decreases at Step 5 (d). In each iteration at Step 7, we have

$\hat{z}_P \leq z_0$ and $\hat{z}_P \leq z_P$. Since z_0 and z_p decrease or remain the same, therefore, the decreased components of $\hat{z}_P$ can never increase.

(vi) If $\hat{z}_P(i,j)$ is 0 at Step 5 then $(i,j) \notin R(i_0)$. (2.33) gives $p_{ij}(\varepsilon) = p_{ij}$ and by (v), p_{ij} must be the initial value defined by (2.10). This proves the assertion.

(vii) This follows from the fact that $\varepsilon \leq \varepsilon_1$ and by (2.26). □

Lemma 2.8. *In each iteration of* `Job_Allocation`*, if there exists a matching X in the bipartite graph $(\eta, Q; \Gamma)$ at Step 1 then (S_j, z_Q, p) satisfies* (BC) *for all $j \in Q$, where z_Q is the vector updated at Step 1.*

Proof. We prove it by induction. In the first iteration at Step 1, the assertion obviously holds. Suppose that there exists a matching X in the bipartite graph $(\eta, Q; \Gamma)$ at Step 1 in the t-th iteration, $t \geq 2$, such that (S_j, z_Q, p) satisfies (BC) for all $j \in Q$, where z_Q is the vector updated at Step 1. Also, for convenience, we denote the vectors/sets calculated for $p(\varepsilon)$ at Step 7 by $z_P(\varepsilon), z_0(\varepsilon), \tilde{z}_P(\varepsilon), \hat{z}_P(\varepsilon), \eta_j(\varepsilon)$, for all $j \in Q$, $\eta(\varepsilon)$ and $\Gamma(\varepsilon)$.

Let $S \in 2^P$ and $j' \in Q$ is such that $\chi_S(i) \leq \hat{z}_P(\varepsilon)(i,j')$ for all $i \in P$ and $|S_{j'}| \leq |S|$. Suppose that $(S, z_Q, p(\varepsilon))$ satisfies (BC) for j'. To prove the assertion, it is equivalent to show that $(S, z_Q(\varepsilon), p(\varepsilon))$ satisfies (BC) for j', where $z_Q(\varepsilon)$ is defined by

$$z_Q(\varepsilon)(i,j) := \begin{cases} 1 & \text{if } j = j' \text{ and } [p_{ij}(\varepsilon) < p_{ij} \text{ or } i \in S] \\ z_Q(i,j) & \text{otherwise} \end{cases} \qquad (\forall (i,j) \in E).$$

If $p_{ij'}(\varepsilon) = p_{ij'}$, for all $(i,j') \in E$, then the following holds:

$$[z_Q(i,j') = 0, \quad z_Q(\varepsilon)(i,j') = 1] \Longrightarrow i \in S.$$

In this case, it is obvious to see that $(S, z_Q(\varepsilon), p(\varepsilon))$ satisfies (BC) for j'. If there exists $(i,j') \in E$ such that $p_{ij'}(\varepsilon) < p_{ij'}$ then

$$|S| = |S_{j'}| = \mu(j') \tag{2.50}$$

by Lemma 2.7 (vii). In this case, firstly, we show that $r^S_{j'}(\varepsilon) \geq r^{S_{j'}}_{j'}(\varepsilon)$, where $r^S_{j'}(\varepsilon)$ and $r^{S_{j'}}_{j'}(\varepsilon)$ are calculated by (2.15) for $p(\varepsilon)$. On contrary, suppose that $r^S_{j'}(\varepsilon) < r^{S_{j'}}_{j'}(\varepsilon)$. Since $\varepsilon \leq \varepsilon_5$, we have

$$\nu_{j'i}(-p_{ij'}(\varepsilon)) \geq r^{S_{j'}}_{j'}(\varepsilon) \quad (\forall i \in S_{j'}). \tag{2.51}$$

Let $i' \in S$ is such that $r^S_{j'}(\varepsilon) = \nu_{j'i'}(-p_{i'j'}(\varepsilon))$. From (2.51), we get $i' \notin S_{j'}$. Then (2.50) yields that there exists $\tilde{i} \in S_{j'}$ such that $\tilde{i} \notin S$. Definition (2.25) gives $z_Q(\tilde{i}, j') = 1$ and (2.51) yields $\nu_{j'\tilde{i}}(-p_{\tilde{i}j'}(\varepsilon)) \geq r^{S_{j'}}_{j'}(\varepsilon)$. But $r^S_{j'}(\varepsilon) < r^{S_{j'}}_{j'}(\varepsilon)$, that is, $\nu_{j'\tilde{i}}(-p_{\tilde{i}j'}(\varepsilon)) > r^S_{j'}(\varepsilon)$. This contradicts that $(S, z_Q, p(\varepsilon))$ satisfies (BC) for j'. Therefore $r^S_{j'}(\varepsilon) \geq r^{S_{j'}}_{j'}(\varepsilon)$. Next, we prove that $(S, z_Q(\varepsilon), p(\varepsilon))$ satisfies (BC) for j'. It suffices to prove (a) of (BC).

Let $i \notin S$ such that $z_Q(\varepsilon)(i, j') = 1$. On contrary, suppose that

$$\nu_{j'i}(-p_{ij'}(\varepsilon)) > r^S_{j'}(\varepsilon). \tag{2.52}$$

Since $(S, z_Q, p(\varepsilon))$ satisfies (BC) for j', we must have $z_Q(i, j') = 0$. This implies that $i \notin S_{j'}$ and $p_{ij'}(\varepsilon) < p_{ij'}$, that is, $(i, j') \in R(i_0)$. By Lemma 2.3, we have $\nu_{j'i}(-p_{ij'}) \leq r_{j'}$. But $r_{j'} \leq r^{S_{j'}}_{j'}$ by definition. Lemma 2.5 and the definition of ε_6 give

$$\nu_{j'i}(-p_{ij'}(\varepsilon)) \leq r^{S_{j'}}_{j'}(\varepsilon). \tag{2.53}$$

Since $r^{S_{j'}}_{j'}(\varepsilon) \leq r^S_{j'}(\varepsilon)$, the inequalities (2.52) and (2.53) contradict. Hence $(S, z_Q(\varepsilon), p(\varepsilon))$ satisfies (BC) for j'. □

Lemma 2.9. *In each iteration of* `Job_Allocation`*, there exists a matching at Step 1 in the bipartite graph* $(\eta, Q; \Gamma)$ *satisfying* (2.21)−(2.23).

Proof. The initial selection of η_j, for all $j \in Q$, by (2.18) implies that there exists a matching at Step 1 in the first iteration satisfying (2.21). Hence, one can find a matching satisfying (2.21)−(2.23). We suppose that there exists a matching $X = \{(S_j, j) \mid j \in Q\}$ in the bipartite graph $(\eta, Q; \Gamma)$ at Step 1 satisfying (2.21)−(2.23) in the t-th iteration, $t \geq 2$. To avoid any confusion, we specify that z_Q is the vector after update at Step 1. The vectors/sets calculated for $p(\varepsilon)$ at Step 7 are denoted by $z_P(\varepsilon), z_0(\varepsilon), \tilde{z}_P(\varepsilon), \hat{z}_P(\varepsilon), \eta_j(\varepsilon)$, for all $j \in Q$, $\eta(\varepsilon)$ and $\Gamma(\varepsilon)$.

Claim: For all $j \in Q$, $(S_j, z_Q, p(\varepsilon))$ satisfies (BC).

[Proof of Claim] Let $(i, j) \in E$ is such that $i \notin S_j$ and $z_Q(i, j) = 1$. (S_j, z_Q, p) satisfies (BC) by Lemma 2.8, hence $\nu_{ji}(-p_{ij}) \leq r_j^{S_j}$. Since $\varepsilon \leq \min\{\varepsilon_5, \varepsilon_6\}$ and by Lemma 2.5, we get

$$\nu_{ji}(-p_{ij}(\varepsilon)) \leq r_j^{S_j}(\varepsilon), \tag{2.54}$$

where $r_j^{S_j}(\varepsilon)$ is calculated by (2.15) for $p(\varepsilon)$. Now let $\nu_{ji}(-p_{ij}(\varepsilon)) > 0$. If $\nu_{ji}(-p_{ij}) > 0$ then $|S_j| = \mu(j)$. If $\nu_{ji}(-p_{ij}) \leq 0$ then $p_{ij}(\varepsilon) < p_{ij}$, that is, $(i, j) \in R(i_0)$, and Lemma 2.7 (vii) implies that $|S_j| = \mu(j)$. This together with (2.54) implies that $(S_j, z_Q, p(\varepsilon))$ satisfies (BC). [end of proof of Claim]

We first consider the case when [Step 5 (a) → Step 7] is executed. Set $\widehat{S}_{\hat{k}} = S_{\hat{k}} \cup \{\hat{i}\}$. Then by Lemma 2.7 (ii), we have $\chi_{\widehat{S}_{\hat{k}}}(i) \leq \hat{z}_P(\varepsilon)(i, \hat{k})$ for all $i \in P$. We show that $\widehat{S}_{\hat{k}} \in \eta_{\hat{k}}(\varepsilon)$. By the above Claim, $(S_{\hat{k}}, z_Q, p(\varepsilon))$ satisfies (BC) for $\hat{k}$. Then obviously $(\widehat{S}_{\hat{k}}, z_Q, p(\varepsilon))$ satisfies (BC) for $\hat{k}$. Also $\tilde{\mu}(\varepsilon)(\hat{k}) \leq |\widehat{S}_{\hat{k}}| \leq \mu(\hat{k})$. Hence $\widehat{S}_{\hat{k}} \in \eta_{\hat{k}}(\varepsilon)$.

Now if $\hat{i} = i_0$ then $\widehat{X} = \{(S_j, j) \mid j \in Q \setminus \{\hat{k}\}\} \cup \{(\widehat{S}_{\hat{k}}, \hat{k})\}$ is a matching satisfying (2.21). Suppose $\hat{i} \neq i_0$ and, without loss of generality, assume that $\hat{i} \in S_{\hat{j}}$. Then there exists a shortest path S from i_0 to $(\hat{i}, \hat{j})$ in T denoted by

$$S = (i_0, j_0), (i_1, j_0), \ldots, (i_s, j_s) = (i_s, \hat{j}), (i_{s+1}, \hat{j}) = (\hat{i}, \hat{j}) \tag{2.55}$$

such that $((i_h, j_h), (i_{h+1}, j_h)) \in A_1$ for $h = 0, 1, \ldots, s-1$ and $((i_h, j_{h-1}), (i_h, j_h)) \in A_2$ for $h = 1, \ldots, s$. Then by the construction of the graph T, $i_h \notin S_{j_h}$ and $i_{h+1} \in S_{j_h}$ for all $h = 0, 1, \ldots, s$. Now define

$$\widehat{S}_{j_h} = \{S_{j_h} \setminus \{i_{h+1}\}\} \cup \{i_h\} \quad (\forall h = 0, 1, \ldots, s). \tag{2.56}$$

Also for any $i \in P$, Lemma 2.7 (ii) gives $\chi_{\widehat{S}_{j_h}}(i) \leq \hat{z}_P(\varepsilon)(i, j_h)$ for all $h = 0, 1, \ldots, s$. We prove that $(\widehat{S}_{j_h}, z_Q, p(\varepsilon))$ satisfies (BC), for all $h = 0, 1, \ldots, s$. From the above Claim, $(S_{j_h}, z_Q, p(\varepsilon))$ satisfies (BC). Since $((i_h, j_h), (i_{h+1}, j_h))$, for all $h = 0, 1, \ldots, s$, lies on the shortest path from i_0 to $(\hat{\imath}, \hat{\jmath})$, by Lemma 2.5 and the fact that $\varepsilon \leq \min\{\varepsilon_5, \varepsilon_6\}$, we have

$$r_{j_h}^{\widehat{S}_{j_h}}(\varepsilon) = \nu_{j_h i_h}(-p_{i_h j_h}(\varepsilon)) = \nu_{j_h i_{h+1}}(-p_{i_h j_{h+1}}(\varepsilon)) = r_{j_h}^{S_{j_h}}(\varepsilon) \quad (\forall h = 0, 1, \ldots, s),$$

which shows that $(\widehat{S}_{j_h}, z_Q, p(\varepsilon))$ satisfies (BC) for $j_h, h = 0, 1, \ldots, s$. Let $\widehat{Q} = \{j_h \mid h = 0, 1, \ldots, s\} \cup \{\hat{k}\}$. Then note that $\widehat{X} = \{(S_j, j) \mid j \in Q \backslash \widehat{Q}\} \cup \{(\widehat{S}_j, j) \mid j \in \widehat{Q}\}$ is a matching in the bipartite graph $(\eta(\varepsilon), Q; \Gamma(\varepsilon))$ satisfying (2.21) and hence one can find a matching at Step 1 in $(t+1)$-th iteration satisfying (2.21)$-$(2.23). The case when [Step 5 (b) $\rightarrow$ Step 7] or [Step 5 (e) $\rightarrow$ Step 7] is executed, Lemma 2.7 and the above Claim guarantee that X is a matching in the bipartite graph $(\eta(\varepsilon), Q; \Gamma(\varepsilon))$ satisfying (2.21). Next, we deal the case when [Step 5 (c) $\rightarrow$ Step 7] is executed. By Lemma 2.4 and since $\varepsilon = \varepsilon_3 < \min\{\varepsilon_1, \varepsilon_2\}$, we get

$$\chi_{S_j}(i) \leq \hat{z}_P(\varepsilon)(i, j) \quad (\forall (i, j) \in E \text{ with } i \neq \hat{\imath}). \tag{2.57}$$

If $\hat{\imath} = i_0$ then from (2.57) and by the above Claim, obviously X is a matching in the bipartite graph $(\eta(\varepsilon), Q; \Gamma(\varepsilon))$ satisfying (2.21). If $\hat{\imath} \neq i_0$ then by the modification at Step 5 (c), we get $\hat{z}_P(\varepsilon)(\hat{\imath}, j) = 0$ for all $j \in Q$. Obviously, X is not a matching in the graph $(\eta(\varepsilon), Q; \Gamma(\varepsilon))$ and we need some manipulation. Since there exists $j \in Q$ such that $\hat{\imath} \in S_j$, without loss of generality, we assume that $\hat{\imath} \in S_{\hat{\jmath}}$. Then there exists a shortest path S from i_0 to $(\hat{\imath}, \hat{\jmath})$ in the graph T denoted by (2.55). Defining the sets $\widehat{S}_{j_h}$, for all $h = 0, 1, \ldots, s$, by (2.56), analogously we can show that $(\widehat{S}_{j_h}, z_Q, p(\varepsilon))$ satisfies (BC) for $j_h, h = 0, 1, \ldots, s$. Let $\widehat{Q} = \{j_h \mid h = 0, 1, \ldots, s\}$ and set

$$\widehat{X} = \{(S_j, j) \mid j \in Q \setminus \widehat{Q}\} \cup \{(\widehat{S}_j, j) \mid j \in \widehat{Q}\}.$$

Observe that $\widehat{X}$ is a matching in the graph $(\eta(\varepsilon), Q; \Gamma(\varepsilon))$ satisfying (2.21).

Finally, we consider the case when [Step 5 (d) $\rightarrow$ Step 7] is executed. Lemma 2.4 and the inequality $\varepsilon = \varepsilon_4 < \{\varepsilon_1, \varepsilon_2, \varepsilon_3\}$ yield (2.57). Now if $\hat{i} \notin S_{\hat{j}}$ then inequality (2.57) holds for all $(i, j) \in E$. Above Claim implies that X is a matching in the graph $(\eta(\varepsilon), Q; \Gamma(\varepsilon))$. If $\hat{i} \in S_{\hat{j}}$ then just like the previous case, we can find a matching $\widehat{X}$ in the bipartite graph $(\eta(\varepsilon), Q; \Gamma(\varepsilon))$ satisfying (2.21). This completes the proof. □

Lemma 2.10. *In each iteration of* `Job_Allocation`*, the quadruple* $(X; p, z_P, z_Q)$ *at Step 1 satisfies* ($ps'1_w$), ($ps'2$)$-$($ps'4$).

Proof. Since in each iteration, $\varepsilon \le \varepsilon_3$ and for any $j \in Q$, $\chi_{S_j}(i) \le \hat{z}_P(i, j)$ for all $i \in P$. Therefore, the definition of $\hat{z}_P$ implies that ($\mathrm{ps}'1_w$) holds. It is obvious to see that if (S_j, z_Q, p), for all $j \in Q$, satisfies (BC) then ($\mathrm{ps}'2$) holds, where z_Q is updated at Step 1. By the Lemma 2.8, (S_j, z_Q, p) satisfies (BC) for all $j \in Q$ at Step 1, therefore ($\mathrm{ps}'2$) always holds at Step 1. As discussed earlier that the initial selection of z_P and z_Q by (2.11) and (2.25) implies that ($\mathrm{ps}'3$) and ($\mathrm{ps}'4$) hold. Now z_P decreases or remains the same and z_Q increases or remains the same in each iteration by Lemma 2.7. Whenever some component of z_P is decreased at Step 5 (d), the corresponding components of z_Q and p are 1 and the lower bound, respectively. Therefore, ($\mathrm{ps}'3$) and ($\mathrm{ps}'4$) hold in each iteration. □

Next is our main result which shows that if `Job_Allocation` terminates then we get a pairwise stable job allocation.

Theorem 2.11. *If* `Job_Allocation` *terminates then the output* $(X; p, z_P, z_Q)$ *satisfies* ($ps'1$)$-$($ps'4$) *and hence* X *is pairwise stable.*

Proof. By Theorem 2.1, X is pairwise stable if $(X; p, z_P, z_Q)$ satisfies ($\mathrm{ps}'1$)$-$($\mathrm{ps}'4$). By Lemma 2.10, ($\mathrm{ps}'1_w$), ($\mathrm{ps}'2$)$-$($\mathrm{ps}'4$) are satisfied at Step 1 in each iteration. If

`Job_Allocation` terminates at Step 2, we observe that $P = S_P$ or for all $(i, j) \in E$ with $i \in P \setminus S_P$, we have $\tilde{z}_P(i, j) = 0$. This means that $(\text{ps}'1_w)$ and $(\text{ps}'1)$ coincide at termination, that is, $(X; p, z_P, z_Q)$ satisfies $(\text{ps}'1)$–$(\text{ps}'4)$. □

In the rest of the work, we shall prove that the `Job_Allocation` terminates after a finite number of iterations.

Let S, $S' \in 2^P$ such that $S \cap S' = \emptyset$ and $|S| = |S'| \neq 0$. We say that $j \in Q$ *replaces* S by S' in the t-th iteration of `Job_Allocation`, $t \geq 2$, if

(i) $S \subseteq S_j$ and $S' \cap S_j = \emptyset$ in $(t-1)$-th iteration.

(ii) $S' \subseteq S_j$ and $S \cap S_j = \emptyset$ in t-th iteration.

For each $j \in Q$, we define $\rho(j) \subseteq P$ in each iteration of `Job_Allocation` at Step 1 by

$$\rho(j) := \{i \in S_j \mid \nu_{ji}(-p_{ij}) = r_j^{S_j}\}.$$

Lemma 2.12. *In each iteration of* `Job_Allocation`*, if $|\rho(j)|$ decreases for some $j \in Q$ then $\tilde{\mu}(j)$ increases or $z_Q(i, j)$ increases for some $i \in P$, where z_Q is the vector updated at Step 1.*

Proof. If $|\rho(j)|$ decreases for some $j \in Q$ at Step 1 in t-th iteration, $t \geq 2$, then either $\tilde{\mu}(j)$ increases or there exists $S, S' \in 2^P$ with $S \cap S' = \emptyset$ and $|S| = |S'| \neq 0$ such that j replaces S by S'. In the later case, one can easily see that z_Q increases. This completes the proof. □

Lemma 2.13. `Job_Allocation` *terminates in a finite number of iterations.*

Proof. We first mention that in each iteration at Step 4 and Step 6, we use Moore-Bellman-Ford algorithm to find the shortest distances from a single source to all other vertices of the graph T. We show that all executions of `Job_Allocation` for different values of ε are finite.

If [Step 5 (a) $\rightarrow$ Step 7] is executed then $\tilde{\mu}$ increases. By Lemma 2.7 (i), $\tilde{\mu}$ increases or remains the same in each iteration. Therefore this execution is possible at most $\sum_{j\in Q}\mu(j)$ times.

If [Step 5 (b) $\rightarrow$ Step 7] is executed then $\hat{z}_P$ increases by Lemma 2.7 (ii). From Lemma 2.7 (v), we observe that [Step 5 (b) $\rightarrow$ Step 7] can be executed at most $|E|$ times.

By Lemma 2.7 (i), z_P and z_0 decrease or remain the same in each iteration and if [Step 5 (c) $\rightarrow$ Step 7] is executed then z_0 decreases. Therefore [Step 5 (c) $\rightarrow$ Step 7] is executed atmost $|P|$ times. Similarly, if [Step 5 (d) $\rightarrow$ Step 7] is executed then z_P decreases. Therefore [Step 5 (d) $\rightarrow$ Step 7] is executed at most $|E|$ times.

Note that in any iteration, if $\sum_{j\in Q}|\rho(j)| = |S_P|$ at Step 1 then [Step 5 (e) $\rightarrow$ Step 7] cannot be executed. For any $j \in Q$, $|\rho(j)|$ may increase, decrease or remain unchanged in any iteration. Lemma 2.12 implies that if $|\rho(j)|$, for some $j \in Q$, decreases then $\tilde{\mu}$ or z_Q increase. By Lemma 2.7 (i), $\tilde{\mu}$ and z_Q remain the same or increase. Therefore, the total number of possible iterations when $|\rho(j)|$, for some $j \in Q$, decreases are $\sum_{j\in Q}\mu(j) + |E|$. Hence [Step 5 (e) $\rightarrow$ Step 7] can be executed at most $2\sum_{j\in Q}\mu(j) + |E|$ times.

Next, we consider the case when [Step 5 (f) $\rightarrow$ Step 7] is executed. We suppose that the above mentioned cases do not occur. Then the sum in (2.22) remains the same or increases. Since P and Q are finite, this sum can be increased a finite number of times only. If the sum in (2.22) remains the same then the sum in (2.23) increases. Again, since P and Q are finite, the sum in (2.23) can be increased a finite number of times if the sum in (2.22) remains same constantly. Therefore, [Step 5 (f) $\rightarrow$ Step 7] can be executed only a finite number of times.

Finally, we see that [Step 5 (f) $\rightarrow$ Step 6] is executed in a finite number of times if the other cases do not occur. Let us suppose the other cases do not occur.

Then in execution of [Step 5 (f) $\to$ Step 6], $R(i_0)$ enlarges or remains the same. Since $R(i_0)$ can be enlarged at most $|E|$ times, we discuss the case when $R(i_0)$ remains the same. In such a case, distance of some $(i, j) \in R(i_0)$ is decreased. Also, the distance of each element of $R(i_0)$ remains the same or decreases in each execution of [Step 5 (f) $\to$ Step 6]. Since finite number of paths from i_0 to each $(i, j) \in R(i_0)$ can be found, therefore, [Step 5 (f) $\to$ Step 6] is executed finite number of times if the other cases do not occur. By Lemma 2.7 (iv), observe that the graph $(\eta, Q; \Gamma)$ remains intact and X is a matching in $(\eta, Q; \Gamma)$ satisfying (2.21)–(2.23).

Thus, `Job_Allocation` terminates after a finite number of iterations. $\square$

Chapter 3

Buyer-seller market with indivisible goods and money

3.1 Introduction

In this chapter, we consider an auction market in which some traders own goods which they want to sell, and some traders have money and they want to utilize it by purchasing goods. Here, the traders who own goods are called *sellers* while the traders having money are called *buyers*. Each seller can trade with at most one buyer and vice versa. Obviously, this markets is a one-to-one matching market. The motivation of the work done in this chapter, is the model of Farooq [15]. In the one-to-one matching model of Farooq [15], preferences of participants are given by strictly increasing linear functions of money. For such linear valuations an algorithm for the existence of pairwise stability is presented in Farooq's model [15]. In [15], money is considered as a continuous variable. Unlike the model of Farooq [15], we will use money as a discrete variable rather than as a continuous variable. Considering money as a discrete variable is useful in such auction markets where the market condition says that each bid should increase the price of the item by, say, 1 USD. Such conditions are empirical since very small increments in the price may not be worthwhile for the auctioneer. Also, by considering money from the set of real numbers includes the possibility of dealing with irrational numbers which is not usually done in real life. Thus by considering money as a discrete variable excludes such a possibility and also it makes the process of money transaction very convenient.

In mathematical economics markets with indivisible goods have been widely studied. Kelso and Crawford [27] presented a two-sided matching model with money and introduced a Gross Substitutes condition. Under Gross Substitutes they showed the non-emptiness of the core. Quinzii [35] considered a model of an exchange economy with two kinds of goods. The first kind of good is perfectly divisible (money) and the other is indivisible. Each participant has a certain amount of money and at most one indivisible good. She proved that the core of the economy is non-empty. Gale [24] considered the model of Quinzii [35] and gave a direct proof of the existence of equilibrium by using a generalization of the lemma of Knaster, Kuratowski and Mazurkewicz [28] in combinatorial topology. Even though we are considering the valuations functions similar to that which were considered by Farooq [15] but we can not adopt his procedure. The procedure presented by Farooq [15] works when money is considered as a continuous variable. Therefore, we develop a different algorithm to show that a pairwise stable outcome always exist for our model. The marriage model by Gale and Shapley [23] is a special case of our model.

We organize this chapter as follows. We describe our model in Section 3.2. We give the definitions of a blocking pair and pairwise stability for this model in Section 3.3. In Section 3.4, we describe an algorithm for stable outcome. Correctness and analysis of the algorithm is shown in Section 3.5.

3.2 Description of seller-buyer model

We consider a model with two kinds of participants and two kinds of goods. The first kind of participants are sellers and each seller owns at most one indivisible good. The other kind of participants are buyers each of whom has a certain amount of money. We also assume that each participant can trade with at most one participant of the opposite side. It is also assumed that the transfer of money from buyer to seller has an upper bound and a lower bound and the money is given in integers.

Mathematically, we describe our model as follows. Denote the finite sets of the sellers and buyers by P and Q, respectively, and the set of all possible seller-buyer pairs by $E = P \times Q$. We express lower and upper bounds of prices by two vectors $\underline{\pi}, \overline{\pi} \in \mathbb{Z}^E$ where $\underline{\pi}_{ij} \leq \overline{\pi}_{ij}$ for each $(i, j) \in E$.[1] A vector $p = (p_{ij} \in \mathbb{Z} \mid (i, j) \in E)$ is called a *price vector* if $\underline{\pi}_{ij} \leq p_{ij} \leq \overline{\pi}_{ij}$ for each $(i, j) \in E$.

It is natural to think that each participant has preferences over the participants of the opposite side. Note here that due to the nature of our model, we depart from our former terminology of saying ν_{ij} and ν_{ji} as valuations. We give the preferences of the participants by utility functions defined below. For each $(i, j) \in E$, $\nu_{ij}(x)$ denotes the utility to the seller i if he/she trades with the buyer j and receives an amount x of money. Similarly, $\nu_{ji}(-x)$ denotes the utility to the buyer j if he/she trades with the seller i and pays an amount x of money. For convenience, we do not write a plus sign with x in $\nu_{ij}(x)$. However, it always means that i is a payee. The negative sign in $\nu_{ji}(-x)$ means j is a payer. Furthermore, for each $(i, j) \in E$, we suppose that ν_{ij} and ν_{ji} are increasing and defined by

$$\nu_{ij}(x) = \alpha_{ij}x + \beta_{ij}, \quad \nu_{ji}(-x) = -\alpha_{ji}x + \beta_{ji}, \tag{3.1}$$

[1] Throughout this chapter, $\mathbb{Z}$ stands for the set of integers. The notation $\mathbb{Z}^E$ stands for the integer lattice whose points are indexed by E.

where α_{ij} and α_{ji} are given positive real numbers, β_{ij} and β_{ji} are any given real numbers and $x \in \mathbb{Z}$.

If $\nu_{ij}(x) \geq 0$ we shall say that the buyer j is *acceptable* to the seller i at x. This means that i is willing to trade with j at the amount x. Similarly, $\nu_{ji}(-x) \geq 0$ means that the seller i is *acceptable* to the buyer j at x. We remark that even if i and j are mutually acceptable they may not be matched with each other since both i and j have preference lists and a stable matching depends upon the preferences. We say that i *prefers* j to j' at x and x' if $\nu_{ij}(x) > \nu_{ij'}(x')$. Similarly, j *prefers* i to i' at x and x' if $\nu_{ji}(-x) > \nu_{ji'}(-x')$. A seller i is *indifferent* between j and j' at x and x' if $\nu_{ij}(x) = \nu_{ij'}(x')$. Moreover, a buyer j is said to be *indifferent* between i and i' at x and x' if $\nu_{ji}(-x) = \nu_{ji'}(-x')$. If $\nu_{ij}(x) \geq 0$ for some $x \in \mathbb{Z}$, then the buyer j is acceptable to the seller i at x by definition. However,if $\nu_{ij}(x) = 0$ then we say that the seller i is indifferent between the buyer j and himself at x. As mentioned earlier that if a participant is unmatched, we say it self-matched. Similarly, if $\nu_{ji}(-x) = 0$ for some $x \in \mathbb{Z}$, then we say that the buyer j is indifferent between the seller i and himself at x. Since the preferences of the participants purely based on the monetary transfer, we can assume that preferences of the participants are not strict.

3.3 Pairwise stability

We know that a matching is said to be pairwise stable if it is individually rational and is not blocked by any seller-buyer pair.

A 4-tuple $(X; p, q, r)$ is said to be an *outcome* if X is a matching, p is a price vector and $(q,r) \in \mathbb{R}^P \times \mathbb{R}^Q$ is defined by

$$q_i = \begin{cases} \nu_{ij}(p_{ij}) & \text{if } (i,j) \in X \text{ for some } j \in Q \\ 0 & \text{otherwise} \end{cases} \qquad (i \in P), \tag{3.2}$$

$$r_j = \begin{cases} \nu_{ji}(-p_{ij}) & \text{if } (i,j) \in X \text{ for some } i \in P \\ 0 & \text{otherwise} \end{cases} \qquad (j \in Q). \tag{3.3}$$

Mathematically, we define the pairwise stability as follows. An outcome $(X;p,q,r)$ is *pairwise stable* if the following two conditions are satisfied:

(ps'1) $q \geq \mathbf{0}$ and $r \geq \mathbf{0}$,

(ps'2) $\nu_{ij}(c) \leq q_i$ or $\nu_{ji}(-c) \leq r_j$ for all $c \in [\underline{\pi}_{ij}, \overline{\pi}_{ij}]_{\mathbb{Z}}$ and for all $(i,j) \in E$.[2]

Condition (ps'1) says that the matching X is individually rational. Condition (ps'2) means $(X;p,q,r)$ is not blocked by any seller-buyer pair. A matching X is called *pairwise stable* if $(X;p,q,r)$ is pairwise stable.

3.4 Existence of pairwise stable outcome

The procedure adopted by Farooq [15] to show the existence of a pairwise stable outcome when money is a continuous variable does not work in our case. We will use different mathematical tools and give a constructive proof to show that there always exists a pairwise stable outcome in the model described in Section 3.2. Initially, we define $p_{ij} \in \mathbb{Z}$, for each $(i,j) \in E$, by[3]

$$p_{ij} = \begin{cases} \overline{\pi}_{ij} & \text{if } \nu_{ji}(-\overline{\pi}_{ij}) \geq 0 \\ \max\left\{\underline{\pi}_{ij}, \left\lfloor \frac{\beta_{ji}}{\alpha_{ji}} \right\rfloor\right\} & \text{otherwise.} \end{cases} \tag{3.4}$$

[2] For any $x, y \in \mathbb{Z}$, we define $[x,y]_{\mathbb{Z}} = \{a \in \mathbb{Z} \mid x \leq a \leq y\}$.

[3] $\lfloor x \rfloor = \sup\{n \in \mathbb{Z} \mid x \geq n\}$.

Then $\underline{\pi}_{ij} \leq p_{ij} \leq \overline{\pi}_{ij}$ for each $(i, j) \in E$. Note that if $\nu_{ji}(-p_{ij}) \geq 0$ for some $(i, j) \in E$, then p_{ij} is the maximum integer in $[\underline{\pi}_{ij}, \overline{\pi}_{ij}]_{\mathbb{Z}}$ for which this inequality holds.

Before describing the algorithm mathematically, we first give an outline of the algorithm. Initially we will exclude all those seller-buyer pairs such that at least one of them is not acceptable to the other. Then, from the set of mutually acceptable seller-buyer pairs, we will find a matching X of all those seller-buyer pairs such that buyer is the most preferred for the seller and the sum of the utilities to the matched buyers is maximum. We define q and r by (3.2) and (3.3), respectively. Then the 4-tuple $(X; p, q, r)$ obviously satisfies (ps'1). However, if there exists a mutually acceptable seller-buyer pair such that the seller is unmatched then it may form a blocking pair, that is, (ps'2) may not hold. To eliminate all such pairs, we will modify the corresponding components of the price vector. It is worthwhile to note that the price vector will be non-increasing and the size of matching will be non-decreasing throughout the algorithm. If the price vector is decreased then the preference lists of the participants may change. Therefore a matched participant may change his partner according to the new preference list. We will make it certain that a matched buyer will remain matched. However, it is not required that a matched seller will remain matched. Throughout our procedure, we will exclude two kinds of unmatched pairs, if they appear; (i) those seller-buyer pairs such that the buyer is not acceptable to the seller and (ii) those seller-buyer pairs such that the corresponding component of the price vector is the lower bound and the payoff of the buyer is not greater than his/her utility to the seller. If the price vector remains unchanged in some iteration, then at least one kind of a seller-buyer pair mentioned above will be eliminated. Since the price vector is discrete and bounded and the number of participants is finite, the algorithm will be terminated after a finite number of iterations.

Now we present our algorithm mathematically. First, we define some subsets of E which will help us to find a matching satisfying (ps'1). Define two subsets L_0 and E_0 of E as follows:

$$L_0 = \{(i,j) \in E \mid \nu_{ji}(-p_{ij}) < 0\}, \tag{3.5}$$

$$E_0 = \{(i,j) \in E \mid \nu_{ij}(p_{ij}) < 0\}. \tag{3.6}$$

Then L_0 is the set of those seller-buyer pairs where the seller is not acceptable to the buyer, whereas, E_0 is the set of those seller-buyer pairs where buyer is not acceptable to the seller. The set of mutually acceptable seller-buyer pairs is defined by

$$\widetilde{E} = E \setminus \{L_0 \cup E_0\}. \tag{3.7}$$

We also define $\tilde{q}_i$, for each $i \in P$, by

$$\tilde{q}_i = \max\{\nu_{ij}(p_{ij}) \mid (i,j) \in \widetilde{E}\} \tag{3.8}$$

and

$$\widetilde{E}_P = \{(i,j) \in \widetilde{E} \mid \nu_{ij}(p_{ij}) = \tilde{q}_i\}. \tag{3.9}$$

The maximum over an empty set is taken to be zero by definition. The set $\widetilde{E}_P$ contains those seller-buyer pairs which are mutually acceptable and the buyer is most preferred for seller out of all acceptable buyers.

Initially we put $r = \mathbf{0}$ and define a subset $\widehat{E}_P$ of $\widetilde{E}_P$ by

$$\widehat{E}_P = \{(i,j) \in \widetilde{E}_P \mid \nu_{ji}(-p_{ij}) \geq r_j\}. \tag{3.10}$$

At this stage, obviously $\widetilde{E}_P = \widehat{E}_P$. However, in the subsequent iterations of the algorithm, $\widehat{E}_P$ may be a proper subset of $\widetilde{E}_P$. Also, define $\widetilde{Q} = \emptyset$. Let X be a matching in the bipartite graph $(P, Q; \widehat{E}_P)$ that satisfies the following conditions:

(a1) X matches all members of $\widetilde{Q}$,

(a2) X maximizes $\sum_{(i,j)\in X} \nu_{ji}(-p_{ij})$ among the matchings satisfying (a1).

Define the vectors q and r by (3.2) and (3.3), respectively. Then from (3.4) and (3.5), we obtain

$$p_{ij} = \underline{\pi}_{ij} \text{ and } \nu_{ji}(-p_{ij}) < r_j \quad ((i,j) \in L_0). \tag{3.11}$$

Let $\widetilde{Q}$ denote the set of matched buyers in X, that is,

$$\widetilde{Q} = \{j \in Q \mid j \text{ is matched in } X\}. \tag{3.12}$$

The main purpose of defining $\widehat{E}_P$ and putting condition (a1) on the matching is to keep the vector r non-decreasing throughout the algorithm. The condition (a2) will help us to prove (ps'2) at termination of the algorithm. Next, define a set U by

$$U = \{(i,j) \in \widetilde{E}_P \mid i \text{ is unmatched in } X\}. \tag{3.13}$$

Then U is the set of all those seller-buyer pairs that are mutually acceptable and the buyer is most preferred for the seller but the seller is unmatched in X.

In each iteration of the algorithm, we will modify p in such a way that (ps'1) and the feasibility of p (that is, $\underline{\pi}_{ij} \le p_{ij} \le \overline{\pi}_{ij}$ for each $(i,j) \in E$) are preserved. To modify p, we find an integer n_{ij}, for each $(i,j) \in U$, by[4]

$$n_{ij} = \max\left\{1, \left\lceil \frac{r_j - \nu_{ji}(-p_{ij})}{\alpha_{ji}} \right\rceil\right\}. \tag{3.14}$$

Define a subset L of U by

$$L = \{(i,j) \in U \mid p_{ij} - n_{ij} < \underline{\pi}_{ij}\}. \tag{3.15}$$

Now we modify the price vector p. The modified price vector is denoted by $\tilde{p}$ and is defined by

$$\tilde{p}_{ij} := \begin{cases} \max\{\underline{\pi}_{ij}, p_{ij} - n_{ij}\} & \text{if } (i,j) \in U \\ p_{ij} & \text{otherwise} \end{cases} \qquad ((i,j) \in E). \tag{3.16}$$

[4] $\lceil x \rceil = \inf\{n \in \mathbb{Z} \mid x \le n\}$.

Then obviously $\underline{\pi}_{ij} \leq \tilde{p}_{ij} \leq \overline{\pi}_{ij}$ for each $(i, j) \in E$. Define a subset $\widetilde{E}_0$ of U as follows:

$$\widetilde{E}_0 := \{(i, j) \in U \mid \nu_{ij}(\tilde{p}_{ij}) < 0\}. \tag{3.17}$$

We finally propose our algorithm.

Algorithm:

Step 0: Put $r = \mathbf{0}$ and $\widetilde{Q} = \emptyset$. Initially define p, L_0, E_0, $\widetilde{E}$, $\tilde{q}$, $\widetilde{E}_P$ and $\widehat{E}_P$ by (3.4)$-$(3.10), respectively. Find a matching X in the bipartite graph $(P, Q; \widehat{E}_P)$ satisfying (a1) and (a2). Define r, $\widetilde{Q}$ and U by (3.3), (3.12) and (3.13), respectively.

Step 1: If $U = \emptyset$ then define q by (3.2) and stop.

Step 2: For each pair $(i, j) \in U$, calculate n_{ij} by (3.14) and find $\tilde{p}$ by (3.16). Define L and $\widetilde{E}_0$ by (3.15) and (3.17), respectively. Update E_0 by $E_0 := E_0 \cup \widetilde{E}_0$ and L_0 by $L_0 := L_0 \cup L$.

Step 3: Put $p := \tilde{p}$ and modify $\widetilde{E}$ by

$$\widetilde{E} := \widetilde{E} \setminus \{L_0 \cup E_0\}. \tag{3.18}$$

Define $\tilde{q}$ by (3.8). Modify $\widetilde{E}_P$ and $\widehat{E}_P$ by (3.9) and (3.10), respectively, for the updated $\widetilde{E}$ and p. Find a matching X in the bipartite graph $(P, Q; \widehat{E}_P)$ satisfying (a1) and (a2). Define r, $\widetilde{Q}$ and U by (3.3), (3.12) and (3.13), respectively. Go to Step 1.

Now we present an example to describe the working of our algorithm.

Example 3.1. We consider two sets $P = \{i_0, i_1, i_2, i_3\}$ and $Q = \{j_0, j_1, j_2, j_3\}$ where P represents the set of buyers and Q represents the set of firms. The set of all possible worker-firm pairs is given by $E = P \times Q$. Define the lower and upper bounds for all $(i, j) \in E$ as follows:

$$\begin{aligned} \underline{\pi}_{ij} &= -1 \qquad \forall (i,j) \in E, \\ \overline{\pi}_{ij_0} &= 4 = \overline{\pi}_{ij_1} \quad \forall i \in P, \\ \overline{\pi}_{ij_2} &= 3 = \overline{\pi}_{ij_3} \quad \forall i \in P. \end{aligned}$$

We assume that valuations are given by (3.1), where $\alpha_{ij}, \alpha_{ji}, \beta_{ij}$ and β_{ji} for each $(i, j) \in E$ are given as follows:

α_{ij}	j_0	j_1	j_2	j_3
i_0	1.5	2	0.5	0.5
i_1	1.5	2	0.5	0.5
i_2	1.5	2	0.5	0.5
i_3	1.5	2	0.5	0.5

Table 3.1: α_{ij} for $(i, j) \in E$

α_{ji}	i_0	i_1	i_2	i_3
j_0	1.5	1.5	1.5	1.5
j_1	2	2	2	2
j_2	0.5	0.5	0.5	0.5
j_3	0.5	0.5	0.5	0.5

Table 3.2: α_{ji} for $(i, j) \in E$

and

β_{ij}	j_0	j_1	j_2	j_3
i_0	1	−3	1.5	10
i_1	8	2	5	−2.5
i_2	4	1	6	5
i_3	2	10	8	3

Table 3.3: β_{ij} for $(i, j) \in E$

β_{ji}	i_0	i_1	i_2	i_3
j_0	−1	12	7	−7
j_1	10	12	20	0
j_2	6	10	3	−0.5
j_3	10	11	12	13

Table 3.4: β_{ji} for $(i, j) \in E$

We start with Step 0 of the algorithm. We set $r = \mathbf{0}$ and $\widetilde{Q} = \emptyset$. We find price vector $p = (p_{i_0j_0}, p_{i_0j_1}, p_{i_0j_2}, p_{i_0j_3}, p_{i_1j_0}, p_{i_1j_1}, \ldots, p_{i_3j_3})$ by (3.4). We have $\nu_{ji}(-\overline{\pi}_{ij}) \geq 0$ for $(i, j) \in E \setminus \{(i_0, j_0), (i_3, j_0), (i_3, j_1), (i_3, j_2)\}$ and $\nu_{ji}(-\overline{\pi}_{ij}) < 0$ for $(i, j) \in \{(i_0, j_0), (i_3, j_0), (i_3, j_1), (i_3, j_2)\}$. Therefore, we have

$$p = (-1, 4, 3, 3, 4, 4, 3, 3, 4, 4, 3, 3, -1, 0, -1, 3).$$

Using the values given in Tables (3.1)−(3.4) and the price vector p, we obtain $\nu_{ij}(p_{ij})$ and $\nu_{ji}(-p_{ij})$ for each $(i, j) \in E$ as follows:

$\nu_{ij}(p_{ij})$	j_0	j_1	j_2	j_3
i_0	−0.5	5	3	11.5
i_1	14	10	6.5	−1
i_2	10	9	7.5	6.5
i_3	0.5	10	7.5	4.5

Table 3.5: $\nu_{ij}(p_{ij})$ for $(i, j) \in E$

$\nu_{ji}(-p_{ij})$	i_0	i_1	i_2	i_3
j_0	0.5	6	1	−5.5
j_1	2	4	12	0
j_2	4.5	8.5	1.5	0
j_3	8.5	9.5	10.5	11.5

Table 3.6: $\nu_{ji}(-p_{ij})$ for $(i, j) \in E$

By (3.5) and (3.6), we have $L_0 = \{(i_3, j_0)\}$ and $E_0 = \{(i_0, j_0), (i_1, j_3)\}$. By (3.7), we find the set of mutually acceptable seller-buyer pairs which is given by:

$$\widetilde{E} = E \setminus \{(i_0, j_0), (i_1, j_3), (i_3, j_0)\}.$$

From (3.8), we find

$$\tilde{q}_{i_0} = 11.5, \quad \tilde{q}_{i_1} = 14, \quad \tilde{q}_{i_2} = 10, \quad \tilde{q}_{i_3} = 10.$$

Using (3.9), we obtain

$$\widetilde{E}_P = \{(i_0, j_3), (i_1, j_0), (i_2, j_0), (i_3, j_1)\}.$$

Since $r = \mathbf{0}$, we have $\widehat{E}_P = \widetilde{E}_P$ by (3.10). The set $\widehat{E}_P$ represents that the firm j_0 has two offers two make a partnerships whereas the firm j_1 and the firm j_3 each has one option to sell its goods. We shall utilize the values given Table (3.6) to find a matching which satisfies (a2). In the bipartite graph $(P, Q; \widehat{E}_P)$, a matching X satisfying conditions (a1) and (a2) comes out to be $\{(i_0, j_3), (i_1, j_0), (i_3, j_1)\}$. We find r by (3.3) and get:

$$r_{j_0} = 6, \quad r_{j_1} = 0, \quad r_{j_2} = 0, \quad r_{j_3} = 8.5. \tag{3.19}$$

Updating $\widetilde{Q}$ by (3.8), we get:

$$\widetilde{Q} = \{j_0, j_1, j_3\}.$$

By (3.13) we have $U = \{(i_2, j_0)\}$. This completes Step 0. We move to Step 2 since $U \neq \emptyset$. Calculating $n_{i_2 j_0}$ by (3.14) we get $n_{i_2 j_0} = 4$. Modifying price vector $\tilde{p}$ by (3.16) we get:

$$\tilde{p} = (-1, 4, 3, 3, 4, 4, 3, 3, 0, 4, 3, 3, -1, 0, -1, 3).$$

For this modified price vector $\tilde{p}$, we have $\nu_{i_2 j_0}(\tilde{p}_{i_2 j_0}) = 4$, $\nu_{j_0 i_2}(-\tilde{p}_{i_2 j_0}) = 7$ and the remaining values given in Table (3.5) and Table (3.6) remain unchanged. For this modified price vector, both L and $\widetilde{E}_0$ are empty. Therefore, E_0 and L_0 remain unchanged for the modified vector $\tilde{p}$.

At Step 3, we modify $\widetilde{E}$ by (3.18). Since both L_0 and E_0 are empty, it follows that $\widetilde{E}$ remains unchanged. By (3.8), we have

$$\tilde{q}_{i_0} = 11.5, \quad \tilde{q}_{i_1} = 14, \quad \tilde{q}_{i_2} = 9, \quad \tilde{q}_{i_3} = 10.$$

Updating $\widetilde{E}_P$ and $\widehat{E}_P$ by (3.9) and (3.10), respectively, we get

$$\begin{aligned} \widetilde{E}_P &= \{(i_0, j_3), (i_1, j_0), (i_2, j_1), (i_3, j_1)\}, \\ \widetilde{E}_P &= \{(i_0, j_3), (i_1, j_0), (i_2, j_1), (i_3, j_1)\}. \end{aligned}$$

In the bipartite graph $(P, Q; \widehat{E}_P)$, a matching

$$X = \{(i_0, j_3), (i_1, j_0), (i_2, j_1)\}$$

satisfies conditions (a1) and (a2). Again, updating r by (3.3), we get

$$r_{j_0} = 6, \quad r_{j_1} = 12, \quad r_{j_2} = 0, \quad r_{j_3} = 8.5. \tag{3.20}$$

The updated $\widetilde{Q}$ and U by (3.8) and (3.13), respectively, are given by

$$\widetilde{Q} = \{j_0, j_1, j_3\}, \qquad U = \{(i_3, j_1)\}.$$

Since $U \neq \emptyset$ we move to Step 2. Now we have $n_{i_3 j_1} = 6$ by (3.14). By (3.16), we have

$$\tilde{p} = (-1, 4, 3, 3, 4, 4, 3, 3, 0, 4, 3, 3, -1, -1, -1, 3).$$

All the values in the Tables (3.5) and (3.5) remain unchanged for this modified price vector $\tilde{p}$ except $\nu_{i_3 j_1}(\tilde{p}_{i_3 j_1}) = 4$ and $\nu_{j_1 i_3}(-\tilde{p}_{i_3 j_1}) = 2$. By (3.15), we have $L = \{(i_3, j_1)\}$. Thus $\widetilde{E} = E \setminus \{(i_0, j_0), (i_1, j_3), (i_3, j_0), (i_3, j_1)\}$. For this updated $\widetilde{E}$, we have

$$\tilde{q}_{i_0} = 11.5, \quad \tilde{q}_{i_1} = 14, \quad \tilde{q}_{i_2} = 9, \quad \tilde{q}_{i_3} = 7.5.$$

Further, by (3.9) and (3.10), we obtain

$$\begin{aligned} \widetilde{E}_P &= \{(i_0, j_3), (i_1, j_0), (i_2, j_1), (i_3, j_2)\}, \\ \widehat{E}_P &= \{(i_0, j_3), (i_1, j_0), (i_2, j_1), (i_3, j_2)\}, \end{aligned}$$

respectively. Finally, we obtain a matching

$$X = \{(i_0, j_3), (i_1, j_0), (i_2, j_1), (i_3, j_2)\}$$

in the bipartite graph $(P, Q; \widehat{E}_P)$ satisfying conditions (a1) and (a2). We see that $U = \emptyset$. Our algorithm terminates at this point ant the matching $X = \{(i_0, j_3), (i_1, j_0), (i_2, j_1), (i_3, j_2)\}$ is a pairwise stable.

3.5 Analysis of the algorithm

In the rest of the work, we present the analysis of the algorithm. For this purpose, we prove some important lemmas related to the algorithm. These lemmas describe important features of the algorithm. These are also very helpful to prove the correctness of the algorithm. We also utilize some of these lemmas to show that the algorithm terminates after a finite number of iterations. We will add prefixes $(old)*$ and $(new)*$ to sets/vectors/integers before and after update, respectively, in any iteration of the algorithm. When the context is clear we will not add these prefixes. We start with the following lemma.

Lemma 3.2. *In each iteration of the algorithm at Step 3, there exists a matching in the bipartite graph* $(P, Q; \widehat{E}_P)$ *satisfying (a1) and (a2).*

Proof. It is enough to prove that $(old)X \subseteq (new)\widehat{E}_P$ in each iteration at Step 3. Initially the set E_0 is defined by (3.6) at Step 0 before finding the matching. Then in each iteration at Step 2, E_0 is augmented if $\widetilde{E}_0$ is nonempty, otherwise, it remains the same. Also $L, \widetilde{E}_0 \subseteq U$ and $U \cap (old)X = \emptyset$ at Step 2. Therefore, (3.16) and (3.18) imply that $(old)X \subseteq (new)\widetilde{E}_P$ at Step 3. By (3.10), $(old)r_j$ is the lower bound of $\nu_{ji}(-(new)p_{ij})$ for each $(i,j) \in (new)\widehat{E}_P$. Therefore, $(old)X \subseteq (new)\widehat{E}_P$. □

The next lemma describes the important features of the algorithm. This lemma will be used to prove the subsequent lemmas. Specifically, the first two parts are crucial in proving the termination of the algorithm in a finite number of iterations.

Lemma 3.3. *In each iteration of the algorithm, the following hold:*

(i) *The price vector* p *decreases or remains the same. In particular, if* $U \setminus \{L \cup \widetilde{E}_0\} \neq \emptyset$ *at Step 2 then* p_{ij} *decreases at Step 3 for all* $(i,j) \in U \setminus \{L \cup \widetilde{E}_0\}$.

(ii) $\widetilde{E}$ *reduces or remains the same. In particular, if* $L \neq \emptyset$ *or* $\widetilde{E}_0 \neq \emptyset$ *at Step 2 then* $\widetilde{E}$ *reduces at Step 3.*

(iii) *The vector* r *increases or remains the same.*

Proof. (i) Initially the price vector p is defined by (3.4) and in each iteration it is modified by (3.16). From (3.16), one can easily see that p decreases or remains the same at Step 3. If $U \neq \emptyset$ then we find $\tilde{p}$ by (3.16) at Step 2. For each

$(i,j) \in U$, n_{ij} is a positive integer. Now, if $U \setminus \{L \cup \widetilde{E}_0\} \neq \emptyset$ then one can easily see from (3.16) that $\tilde{p}_{ij} = p_{ij} - n_{ij}$ for all $(i,j) \in U \setminus \{L \cup \widetilde{E}_0\}$ at Step 2. This proves the assertion.

(ii) Initially $\widetilde{E}$ is defined by (3.7) at Step 0 and it is modified by (3.18) at Step 3 in each iteration. If $L = \widetilde{E}_0 = \emptyset$ at Step 2 then $\widetilde{E}$ remains the same at Step 3. If $L \neq \emptyset$ at Step 2 then $\widetilde{E}$ reduces at Step 3. If $\widetilde{E}_0 \neq \emptyset$ at Step 2 then E_0 enlarges at Step 2 and consequently, $\widetilde{E}$ reduces at Step 3.

(iii) By the proof of Lemma 3.2, we have $(old)X \subseteq (new)\widehat{E}_P$ at Step 3. Hence, for each $j \in (old)\widetilde{Q}$, there exists $(i,j) \in (new)\widehat{E}_P$ such that $\nu_{ji}(-(new)p_{ij}) \geq (old)r_j$. Also, $(old)r_j = 0$ for each $j \in Q \setminus (old)\widetilde{Q}$ by (3.3). Since $(new)X \subseteq (new)\widehat{E}_P$ and by (a1) we have $(old)\widetilde{Q} \subseteq (new)\widetilde{Q}$, it holds that $(new)r_j = \nu_{ji}(-(new)p_{ij}) \geq (old)r_j$ for each $(i,j) \in (new)X$. Moreover, $(new)r_j = (old)r_j = 0$ for each $j \in Q \setminus (new)\widetilde{Q}$. Hence, the vector r increases or remains the same. □

The next two lemmas do hold in each iteration of the algorithm at Step 3.

Lemma 3.4. *In each iteration of the algorithm at Step 3, we have* $\nu_{ji}(-((old)p_{ij} - n_{ij})) \geq (old)r_j$ *for each* $(i,j) \in (old)U$*, where* n_{ij} *is calculated at Step 2. Furthermore, if* $\nu_{ji}\,(-((old)p_{ij} - n_{ij})) > (old)r_j$ *for some* $(i,j) \in (old)U$ *then* $(old)p_{ij} - n_{ij}$ *is the maximum integer for which this inequality holds.*

Proof. Let $(i,j) \in (old)U$ at Step 3. Then

$$\begin{aligned}
\nu_{ji}(-((old)p_{ij} - n_{ij})) &\geq \nu_{ji}(-(old)p_{ij}) + \alpha_{ji} \left\lceil \frac{(old)r_j - \nu_{ji}(-(old)p_{ij})}{\alpha_{ji}} \right\rceil \\
&\geq \nu_{ji}(-(old)p_{ij}) + (old)r_j - \nu_{ji}(-(old)p_{ij}) \\
&= (old)r_j.
\end{aligned}$$

Next we prove that if $\nu_{ji}(-((old)p_{ij} - n_{ij})) > (old)r_j$ then $(old)p_{ij} - n_{ij}$ is the maximum integer for which this holds. Assume that $\nu_{ji}\,(-((old)p_{ij} - n_{ij})) > (old)r_j$. Since $n_{ij} \geq 1$ by definition (3.14), we first consider the case when

$\frac{(old)r_j - \nu_{ji}(-(old)p_{ij})}{\alpha_{ji}} < 1$, that is, $n_{ij} = 1$. By (a2), we obtain $\nu_{ji}(-(old)p_{ij}) \leq (old)r_j$. But $(old)r_j < \nu_{ji}(-((old)p_{ij} - 1))$. This proves the assertion when $\frac{(old)r_j - \nu_{ji}(-(old)p_{ij})}{\alpha_{ji}} < 1$. Next we consider the case when $\frac{(old)r_j - \nu_{ji}(-(old)p_{ij})}{\alpha_{ji}} \geq 1$, that is, $n_{ij} = \left\lceil \frac{(old)r_j - \nu_{ji}(-(old)p_{ij})}{\alpha_{ji}} \right\rceil$. Therefore

$$\begin{aligned}
\nu_{ji}(-((old)p_{ij} - n_{ij} + 1)) &= \nu_{ji}(-(old)p_{ij}) + \alpha_{ji}\left(\left\lceil \frac{(old)r_j - \nu_{ji}(-(old)p_{ij})}{\alpha_{ji}} \right\rceil - 1\right) \\
&< \nu_{ji}(-(old)p_{ij}) + (old)r_j - \nu_{ji}(-(old)p_{ij}) \\
&= (old)r_j.
\end{aligned}$$

This means that $\nu_{ji}(-((old)p_{ij} - n_{ij})) > (old)r_j > \nu_{ji}(-((old)p_{ij} - n_{ij} + 1))$ which implies that $(old)p_{ij} - n_{ij}$ is the maximum integer for which $\nu_{ji}(-((old)p_{ij} - n_{ij})) > (old)r_j$. □

Lemma 3.5. *In each iteration of the algorithm at Step 3, we have* $\nu_{ji}(-(new)p_{ij}) \geq (old)r_j$ *for each* $(i,j) \in (old)U \setminus L$*, where* L *is defined at Step 2. Furthermore, if* $\nu_{ji}(-(new)p_{ij}) > (old)r_j$ *for some* $(i,j) \in (old)U \setminus L$ *then* $(new)p_{ij}$ *is the maximum integer in* $[\underline{\pi}_{ij}, \overline{\pi}_{ij}]_{\mathbb{Z}}$ *for which this inequality holds.*

Proof. We prove the second part of the assertion. Assume that $\nu_{ji}(-(new)p_{ij}) > (old)r_j$ for some $(i,j) \in (old)U \setminus L$ at Step 3. Also by (a2), we obtain $(old)r_j \geq \nu_{ji}(-(old)p_{ij})$. Thus, $\nu_{ji}(-(new)p_{ij}) > \nu_{ji}(-(old)p_{ij})$. Therefore, (3.16) implies that $(new)p_{ij} = (old)p_{ij} - n_{ij}$, where n_{ij} is calculated at Step 2. The second part of the assertion follows from the second part of Lemma 3.4. □

Lemma 3.6. *In each iteration of the algorithm at Step 3, we have* $(new)p_{ij} = \underline{\pi}_{ij}$ *and* $\nu_{ji}(-(new)p_{ij}) \leq (old)r_j$ *for each* $(i,j) \in L$*, where* L *is defined at Step 2.*

Proof. Let $(i,j) \in L$ at Step 3. Then the first part of the assertion is true by (3.15) and (3.16). We prove the second part. Since $(new)p_{ij} > (old)p_{ij} - n_{ij}$, it holds that $\nu_{ji}(-(new)p_{ij}) < \nu_{ji}(-((old)p_{ij} - n_{ij}))$. Also $L \subseteq (old)U$. Thus, Lemma 3.4

implies $\nu_{ji}(-((old)p_{ij}-n_{ij})) \geq (old)r_j$. If $\nu_{ji}(-((old)p_{ij}-n_{ij})) = (old)r_j$ then the result trivially holds. If $\nu_{ji}(-((old)p_{ij}-n_{ij})) > (old)r_j$ then again by Lemma 3.4, $(old)p_{ij} - n_{ij}$ is the maximum integer for which this inequality holds. Since $(new)p_{ij} > (old)p_{ij} - n_{ij}$, it holds that $\nu_{ji}(-(new)p_{ij}) \leq (old)r_j$. □

The next lemma, in some sense, is more general than Lemma 3.5 and will play a critical role in proving (ps'2).

Lemma 3.7. *In each iteration of the algorithm at Step 3, if $\nu_{ji}(-(new)p_{ij}) > (new)r_j$ for some $(i,j) \in E$ then $(new)p_{ij}$ is the maximum integer in $[\underline{\pi}_{ij}, \overline{\pi}_{ij}]_{\mathbb{Z}}$ for which this inequality holds.*

Proof. Assume that $\nu_{ji}(-(new)p_{ij}) > (new)r_j$ for some $(i,j) \in E$ in the first iteration of the algorithm at Step 3. Then $\nu_{ji}(-(new)p_{ij}) > 0$ since $(new)r_j \geq 0$. If $(i,j) \notin (old)U$ then by (3.16), $(new)p_{ij}$ at Step 3 is the initial value defined by (3.4). The definition (3.4) yields that $(new)p_{ij}$ is the maximum integer in $[\underline{\pi}_{ij}, \overline{\pi}_{ij}]_{\mathbb{Z}}$ for which $\nu_{ji}(-(new)p_{ij}) > 0$ at Step 3. Thus, the result holds in this case. Now, let $(i,j) \in (old)U$ at Step 3. Lemma 3.3 (iii) yields that $(new)r_j \geq (old)r_j$. Therefore, $\nu_{ji}(-(new)p_{ij}) > (old)r_j$. Hence, Lemma 3.6 implies that $(i,j) \in (old)U \setminus L$, where L is defined at Step 2. Thus, by Lemma 3.5, we conclude that $(new)p_{ij}$ is the maximum integer in $[\underline{\pi}_{ij}, \overline{\pi}_{ij}]_{\mathbb{Z}}$ for which this inequality holds. This proves the result in the first iteration.

We suppose that the assertion holds in all iterations fewer than t, $t \geq 2$. We shall show that the assertion holds in t-th iteration. Assume that $\nu_{ji}(-(new)p_{ij}) > (new)r_j$ for some $(i,j) \in E$ at Step 3 in t-th iteration of the algorithm. Firstly, consider the case when $(i,j) \notin (old)U$ at Step 3 in t-th iteration. Then $(old)p_{ij} = (new)p_{ij}$ by (3.16). Therefore, by assumption we get

$$\nu_{ji}(-(old)p_{ij}) = \nu_{ji}(-(new)p_{ij}) > (new)r_j. \tag{3.21}$$

Lemma 3.3 (iii) and (3.21) imply that $\nu_{ji}(-(old)p_{ij}) > (old)r_j$. By the induction hypothesis, $(old)p_{ij}$ is the maximum integer in $[\underline{\pi}_{ij}, \overline{\pi}_{ij}]_{\mathbb{Z}}$ for which this inequality holds. Thus, by (3.21), $(new)p_{ij}$ is the maximum integer in $[\underline{\pi}_{ij}, \overline{\pi}_{ij}]_{\mathbb{Z}}$ for which $\nu_{ji}(-(new)p_{ij}) > (new)r_j$. Secondly, we consider the case when $(i,j) \in (old)U$ at Step 3 in t-th iteration. Lemma 3.3 (iii) gives $(new)r_j \geq (old)r_j$. Hence, $\nu_{ji}(-(new)p_{ij}) > (old)r_j$. Therefore, $(i,j) \notin L$ by Lemma 3.6. Thus, by Lemma 3.5, $(new)p_{ij}$ is the maximum integer in $[\underline{\pi}_{ij}, \overline{\pi}_{ij}]_{\mathbb{Z}}$ for which $\nu_{ji}(-(new)p_{ij}) > (old)r_j$. Consequently, $(new)p_{ij}$ is the maximum integer in $[\underline{\pi}_{ij}, \overline{\pi}_{ij}]_{\mathbb{Z}}$ for which $\nu_{ji}(-(new)p_{ij}) > (new)r_j$. □

Theorem 3.8. *If the algorithm terminates then $(X; p, q, r)$ satisfies (ps'1) and (ps'2).*

Proof. Suppose that the algorithm terminates at Step 1. Then $U = \emptyset$ and let $(X; p, q, r)$ be the 4-tuple obtained at termination. Initially we set $r = \mathbf{0}$ and find a matching in the bipartite graph $(P, Q; \widehat{E}_P)$ satisfying (a1) and (a2). Then we define the vector r by (3.3) at Step 0. Also, initially we define $\widetilde{E}$ by (3.7). Therefore

$$\nu_{ij}(p_{ij}) \geq 0 \text{ and } \nu_{ji}(-p_{ij}) \geq 0 \quad ((i,j) \in \widetilde{E}) \tag{3.22}$$

at Step 0. Since in each iteration we modify $\widetilde{E}$ by (3.18) at Step 3, therefore (3.22) holds in each iteration at Step 3. Also $\widehat{E}_P \subseteq \widetilde{E}$ in each iteration, the definitions (3.2) and (3.3) imply $q \geq \mathbf{0}$ and $r \geq \mathbf{0}$ at termination. Thus, (ps'1) holds true.

Next we show that $(X; p, q, r)$ satisfies (ps'2). We claim that for any $(i,j) \in E$, if there exists a $c \in [\underline{\pi}_{ij}, \overline{\pi}_{ij}]_{\mathbb{Z}}$ with $c > p_{ij}$ then the following inequality holds:

$$\nu_{ji}(-c) \leq r_j. \tag{3.23}$$

If the algorithm terminates in the first iteration then by (3.4) the inequality (3.23) obviously holds. Otherwise, we divide our argument in two cases: $\nu_{ji}(-p_{ji}) \leq r_j$

and $\nu_{ji}(-p_{ji}) > r_j$. If $\nu_{ji}(-p_{ji}) \leq r_j$ then the monotonicity of ν_{ji} implies (3.23). If $\nu_{ji}(-p_{ji}) > r_j$ then by Lemma 3.7, we obtain (3.23). Further, since $X \subseteq \widetilde{E}_P$, the definition (3.9) of $\widetilde{E}_P$ implies that for any matched seller i in X, the following inequality is satisfied:

$$\nu_{ij}(p_{ij}) \leq q_i \quad ((i,j) \in E). \tag{3.24}$$

Note that $U = \emptyset$. Also, $L_0 \cup E_0$ and $\widetilde{E}$ are disjoint and cover all of E. Therefore, if there exists some seller i unmatched in X then $(i,j) \in L_0 \cup E_0$ for each $j \in Q$. If $(i,j) \in E_0$ for some $j \in Q$ then

$$\nu_{ij}(p_{ij}) \leq 0 = q_i \tag{3.25}$$

by definition. From (3.24) and (3.25), we obtain

$$\nu_{ij}(p_{ij}) \leq q_i \quad ((i,j) \in E \setminus L_0). \tag{3.26}$$

Moreover, we assert that

$$p_{ij} = \underline{\pi}_{ij} \text{ and } \nu_{ji}(-p_{ij}) \leq r_j \quad ((i,j) \in L_0). \tag{3.27}$$

If the algorithm terminates in the first iteration then (3.27) is true at Step 1 by (3.11). Otherwise, Lemma 3.3 (iii) and Lemma 3.6 imply (3.27).

Thus for any $(i,j) \in E$ and $c \in [\underline{\pi}_{ij}, \overline{\pi}_{ij}]$ with $c \leq p_{ij}$, the inequalities (3.26) and (3.27) imply that

$$\nu_{ij}(c) \leq q_i \text{ or } [\nu_{ji}(-p_{ij}) \leq r_j \text{ and } p_{ij} = \underline{\pi}_{ij}]. \tag{3.28}$$

The inequalities (3.23) and (3.28) imply that (ps'2) is satisfied. □

Theorem 3.9. *The algorithm terminates after a finite number of iterations.*

Proof. In each iteration of the algorithm at Step 2, either L and $\widetilde{E}_0$ are empty or at least one of them is non-empty. We first consider the case when $L = \widetilde{E}_0 = \emptyset$.

Then by Lemma 3.3 (i), p_{ij} decreases for each $(i, j) \in U$. Note that p is discrete and bounded, and by Lemma 3.3 (i), it decreases or remains the same. Therefore, the vector p can be decreased a finite number of times.

Next we consider the case when $L \neq \emptyset$ or $\widetilde{E}_0 \neq \emptyset$. In either case, $\widetilde{E}$ reduces at Step 3 by Lemma 3.3 (ii). Furthermore, $\widetilde{E}$ reduces or remains the same in each iteration of the algorithm by Lemma 3.3 (ii). Therefore, this case is possible at most $|E|$ times. $\square$

Chapter 4

Summary and discussion

In this thesis, we have considered two types of matching models. In these models participants can exchange money. In Chapter 2, a one-to-many job market is studied with linear valuations. These linear valuations are strictly increasing functions of money. In this model, we have treated money as a continuous variable. In Chapter 3, we have considered a one-to-one matching model with valuations defined on money. Unlike the model in Chapter 2, the money in Chapter 3 is considered as a discrete variable. We now discuss the issues relating to these two models (continuous and discrete) in Section 4.1 and 4.2, respectively.

4.1 Money as a continuous variable

In Chapter 2, we have considered a two-sided matching model with linear valuations. These valuations are defined by (2.9). In this model money is considered as a continuous variable. The money involved in this model is possibly bounded, that is, there may be a lower and an upper bound on the money. We have shown that with these assumptions, our model is a common generalization of the marriage model [23], the assignment game [46] and the hybrid models [14, 48]. The assumption of possibly bounded money is very useful in the comparison of our model with well known existing models.

Another important aspect of our model is that it is a one-to-many matching model. That is, each worker can work for at most one firm and each firm can employ as many workers as it wishes. Due to this important feature of our model, the model of Farooq [15] becomes a special case of our model. We have also presented an algorithm to obtain a pairwise stable outcome for our model. We have discussed the important features of our algorithm in Section 2.6, where we have shown that our algorithm works correctly and terminates after a finite number of iterations. We have established Lemma 2.13 to prove that our algorithm terminates after a finite number of iterations. To provide the correctness of our procedure we have proved Theorem 2.11, which states that our algorithm yields a pairwise stable outcome.

4.1.1 List of some open problems related to the model in Chapter 2

Here we list some of the open problems that are related to one-to-many matching model with linear valuations and money as a continuous variable.

- One can see that the complexity of our algorithm may not be polynomial, specially when $\mu \neq (1, \ldots, 1)$. Thus it is an open problem to design an algorithm which finds a stable outcome in our model having a polynomial complexity. One approach may be the use of linear programming theory to make an efficient algorithm.

- In this model, each firm may be allowed to hire as many workers as it wishes and each worker can work for more than one firm. Preferences of participants may be considered in terms of continuous, strictly increasing linear functions. We do not know whether a pairwise stable outcome exists for such a many-to-many matching market or not. Another open problem is to consider a many-to-many matching market with linear valuations.

- It would be interesting to consider non-linear valuations for a matching market and to show the existence of pairwise stability. For convenience, one can consider a one-to-one matching model with preferences of participants given in terms of non-linear valuations and show the existence of pairwise stability for this simpler market. Afterwards an attempt can be made to show the existence of pairwise stable outcome for a many-to-many matching model with non-linear valuations. If one would be able to prove the existence of pairwise stability for such a matching model, it will include a large class of two-sided matching models as special cases. Another interesting direction of work is to consider preferences in terms of non-linear functions.

4.2 Money as a discrete variable

In Chapter 3, we have studied a one-to-one matching model with linear valuations. In this model each buyer can trade with one seller and vice versa. For this two-sided matching we have considered money as a discrete variable. More specifically, the money is given in integers. Preferences of participants purely depend on the monetary transfer, therefore these preferences are not strict. We have presented the definition of pairwise stability for this model. To show the existence of stable matching for this model, we have designed an algorithm in Section 3.4. We have established different results (Theorem 3.8 and Theorem 3.9) that guaranty the existence of a pairwise stable outcome in our model.

4.2.1 List of some open problems related to the model in Chapter 3

- It can be noted from Theorem 3.9 that the complexity of our algorithm may depend on the size of those intervals where prices fall. It would be worthwhile for someone to devise an algorithm whose complexity is polynomial in the number of participants.

- One can generalize this one-to-one market by a model in which the preferences are given in terms of general increasing functions of money. We believe that this problem can be done by using a similar procedure. In such a problem, we need to find an integer similar to n_{ij} given by (3.14), for general increasing functions.

- It would be interesting to consider the generalization of the model in two respects, that is, many-to-many matching model with preferences given in terms of general increasing functions of money. In [7], Crawford and Knoer used the existence of stability in discrete model to show the existence of pairwise stable outcome for continuous model. Therefore, one can use a similar approach to show the existence of stable outcome for a continuous model. That is, one can utilize the existence of pairwise stability for discrete many-to-many matching model with preferences given in terms of general increasing functions to show the existence for continuous case.

4.3 Few more directions

Now we discuss briefly some more directions in which future work could be done for both models.

In many well known models like [14, 15, 18, 19, 42], existence of pairwise stable outcome is shown but structures of outcomes are not discussed. In two-sided matching literature, the existence of stable outcomes is also proven by either linear programming or by a fixed point theorem of an appropriately defined monotonic mapping. In fixed point theoretical approach mostly Tarski's Fixed Point Theorem [50] is used for the existence of stable outcome. The advantage of applying Tarski's Fixed Point Theorem [50] is that the lattice structure of the set of stable outcomes is also obtained. In [2], Adachi characterized the marriage problem [23] as the fixed points of an increasing function. He showed the existence of pairwise stable outcome and lattice property is an immediate consequence of Tarski's fixed point theorem [50]. When preferences are not strict, such a formulation cannot fully characterize the set of stable matchings. In the marriage model [23], the preferences are strict, whereas the preferences in our model are not strict. To study the lattice structure of outcomes and existence without designing an algorithm it would be worthwhile to consider a two-sided matching problem with linear valuations (continuous or discrete) and prove the existence by using fixed point theory.

Farooq *et al.* investigated structures of stable outcomes in the paper [16]. They also considered a very interesting problem where only a matching or a salary vector is given and the aim was to decide whether it could be expanded into a pairwise stable outcome. One can also consider such type of assumption for a one-to-many model to investigate the same problem.

Artemov [3], analyzed the deferred acceptance algorithm when the preferences are known with an error. One can also consider the above models with error in preferences and investigate how this error would effect the stability of outcome and salary vector?

References

[1] A. Abdulkadiroğlu, T. Sönmez: House allocation with existing tenants. *Journal of Economic Theory*, **88** (1999), 233–260.

[2] H. Adachi: On a charatericzation of stable matchings. *Economics Letters*, **68** (2000), 43–49.

[3] G. Artemov: Matching and price competation: would personalized prices help? *International Journal of Game Theory*, **36** (2008), 321–331.

[4] Y. Ali and R. Farooq: Pairwise stability in a two-sided matching market with indivisible goods and money. *Journal of the Operations Research Society of Japan*, **54** (2011), 1–11.

[5] Y. Ali, R. Farooq: Existence of stable outcome for a job market with linear valuations and possibly bounded salaries. *Pacific Journal of Optimization*, **7** (2011).

[6] R. E. Bellman: On a routing problem. *Quarterly of Applied Mathematics*, **16** (1958), 87–90.

[7] V. P. Crawford and E. M. Knoer: Job matching with heterogeneous firms and workers. *Econometrica*, **49** (1981), 437–450.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest: Introduction to Algorithms. *MIT Press and McGraw-Hill*, Second Edition, 2001.

[9] Canadian Resident Matching Service. How the matching algorithm works. Web document available at *http://www.carms.ca/matching/algorith.htm*.

[10] G. B. Dantzig: Linear programming and extensions. *Princeton University Press*, Princeton, 1963.

[11] G. Demange, D. Gale: The strategy structure of two-sided matching markets. *Econometrica*, (53) (1985), 873–888.

[12] G. Demange, D. Gale, M. Sotomayor: Multi-item auctions. *Journal of Political Economy*, **94** (1986), 863–872.

[13] E. W. Dijkstra: A note on two problems in connexion with graphs. *Numerische Mathematik*, (1) (1959), 269–271.

[14] K. Eriksson and J. Karlander: Stable matching in a common generalization of the marriage and assignment models. *Discrete Mathematics*, **217** (2000), 135–156.

[15] R. Farooq: A polynomial-time algorithm for a stable matching problem with linear valuations and bounded side payments. *Japan Journal of Industrial and Applied Mathematics*, **25** (2008), 83–98.

[16] R. Farooq, Y. T. Ikebe and A. Tamura: On labor allocation model with possibly bounded salaries. *Journal of the Operations Research Society of Japan*, **51** (2008), 136–154.

[17] L. R. Ford and D. R. Fulkerson: Flows in networks. *Princeton University Press*, 1962.

[18] S. Fujishige and A. Tamura: A general two-sided matching market with discrete concave utility functions. *Discrete Applied Mathematics*, **154** (2006), 950–970.

[19] S. Fujishige and A. Tamura: A two-sided discrete-concave market with possibly bounded side payments: An approach by discrete convex analysis. *Mathematics of Operations Research*, **32** (2007), 136–155.

[20] S. Fujishige and Z. Yang: A note on Kelso and Crawford's gross substitutes condition. *Mathematics of Operations Research*, **28** (2003), 463–469.

[21] D. Gale: Equilibrium in exchane economy with menoy. *International Journal of Game Theory*, **13** (1983), 61–64.

[22] D. Gusfield, R. W. Irving: The stable marriage problem: structure and algorithms. *MIT Press*, 1989.

[23] D. Gale and L. S. Shapley: College admissions and the stability of marriage. *The American Mathematical Monthly*, **69** (1962), 9–15.

[24] D. Gale: Equilibrium in a discrete exchange economy with money. *International Journal of Game Theory*, **13** (1984), 61–64.

[25] F. Gul and E. Stacchetti: Walrasian equilibrium with gross substitutes. *Journal of Economic Theory*, **87** (1999), 95–124.

[26] M. Kaneko, The central assignment game and the assignment markets, *Journal of Mathematical Economics*, **10** (1982), 205–232.

[27] J. A. S. Kelso and V. P Crawford: Job matching coalition formation, and gross substitution. *Econometrica*, **50** (1982), 1483–1504.

[28] B. Knaster, C. Kuratowski and S. Mazurkiewicz: Ein Beweis des Fixpunktsatzes für n-dimensionale Simplexe. *Fundamenta Mathematicae*, **14** (1929), 132–137.

[29] D. E. Knuth: Marriages stables. *Les Presses de I'Universite' de Montre'al*, Montreal, 1976.

[30] K. Murota, Convexity and Steinitz's exchange property, *Advances in Mathematics*. **124** (1996), 272–311.

[31] K. Murota: Discrete convex analysis. *Mathematical Programming*, **83** (1998), 313–371.

[32] K. Murota: Discrete Convex Analysis. *Society for Industrial and Applied Mathematics*, Philadelphia, 2003.

[33] K. Murota and A. Shioura: M-convex function on generalized polymatroid. *Mathematics of Operations Research*, **24** (1999), 95–105.

[34] National Resident Matching Program. How the matching algorithm works. Web document available at *http://www.nrmp.org/res match/about res/algorithms.html*.

[35] M. Quinzii: Core and competitive equilibria with indivisibilities. *International Journal of Game Theory*, **13** (1984), 41–60.

[36] A. E. Roth: Stability and polarization of interests in job matching. *Econometrica*, **52** (1984), 47–57.

[37] A.E. Roth, A. Postlewaite: Weak versus strong domination in a market with indivisible goods. *Journal of Mathematical Economics*, **4** (1977), 131–137.

[38] A. Roth, U.G. Rothblum, J.H. Vande Vate: Stable matching, optimal assignments and linear programming. *Mathematics of Operations Research*, **18** (1993), 803–828.

[39] A.E. Roth, T. Sönmez, M.U. Ünver: Kidney exchange. *Quarterly Journal of Economics* , **119** (2004), 457–488.

[40] A. E. Roth, T. Sönmez, M. U. Ünver: Pairwise kidney exchange. *Journal of Economic Theory*, **125** (2005), 151–188.

[41] A. Roth, M. Sotomayor: Two-sided matching. A study in game-theoretic modeling and analysis. *Econometric Society Monographs*, 2nd ed., Vol. 18, Cambridge University Press.

[42] A. Roth, M. Sotomayor: Stable outcomes in discrete and continuous models of two-sided matching: A unified treatment. *The Brazilian Review of Econometrics*, **(**16) (1996), 1–24.

[43] U.G. Rothblum: Characterization of stable matchings as extreme points of a polytope. *Mathematical Programming*, **54** (1992), 57–67.

[44] Scottish PRHO Allocations Scheme. The Scottish PRHO Allocation Scheme. Web document available at *http://www.nes.scot.nhs.uk/spa*.

[45] L. Shapley, H. Scarf: On cores and indivisibility. *Journal of Mathematical Economics*, **1** (1974), 23–28.

[46] L. S. Shapley and M. Shubik: The assignment game I: The core. *International Journal of Game Theory*, **1** (1972), 111–130.

[47] A. Schrijver: Combinatorial optimization: polyhedra and efficiency. *Springer*, (2003).

[48] M. Sotomayor: Existence of stable outcomes and the lattice property for a unified matching market. *Mathematical Social Sciences*, **39** (2000), 119–132.

[49] M. Sotomayor: Core structure and comparative statics in a hybrid matching market. *Games and Economic Behavior*, **60** (2007), 357–380.

[50] A. Tarski: A lattice-theoretic fixpoint theorem and its applications. *Pacific Journal of Mathematics*, **5** (1955), 285–309.

[51] J.H. Vande Vate: Linear programming brings marital bliss. *Operations Research Letters*, **8** (1989), 147–153.

[52] D. B. West: Introduction to Graph Theory (2nd ed.). *Prentice Hall*, (1999).