Chapter 3

Efficient Double Auctions under constraints

3.1 Introduction

Double Auction [M1992] and [GD1998] is a widely used auction mechanism. Most of the electronic markets like multi commodity exchanges or financial markets are built up around this mechanism, which involves multiple buyers, sellers, the continuous double auction and a clearinghouse. It is a mechanism for clearing markets with multiple buyers and sellers. In double auction markets, buyers submit their bids and sellers submit asks. A transaction occurs if the buyer’s bid price exceeds ask price of the seller. Two main institutions for double auctions are continuous double auction and clearing house or continuous call double auctions. A continuous double auction is one in which many individual transactions are carried out and trading does not stop. Call markets, on the other hand, are periodic versions of continuous double auctions, where bids from buyers and asks from sellers are collected over a specified interval of time and the market is cleared at the end of interval. The continuous call double auction is the oldest practised type of market for exchange of stocks, where buyers and sellers post their respective bids and asks continuously. Online trading systems based on double auction mechanism have been implemented in many stock exchanges worldwide. Stock Exchanges such as New York, Tokyo [MRS1992] use this mechanism for online trading. In Arizona Stock Exchange [www.azx.com] bids and asks are transparent. In other words, all the other users know it. Most of the electronic markets like multi commodity exchanges or financial markets are built up around the mechanism of double auction that involves multiple buyers, sellers, the continuous double auction and a clearinghouse. Generally, stocks are homogenous goods and buyers do not have preferences over a designated stock. Double auction-based mechanisms have also been widely used in multi commodity stock exchanges like National Stock Exchange of India [MCS2003] and [NSE1993]. In these cases, the commodity is substitutable and buyer does not have any preference for a particular seller. So the supply from different sellers can be aggregated to satisfy the demand from
different buyers. Another feature of these types of auctions is that goods (e.g. stocks, commodities in stock exchanges) do not have any differentiating features apart from price and quantity. In these cases price and quantity become only differentiating factors and matching is done based on them. However in some cases this assumption may not be valid. Even though the commodity in exchange is the same, there may be some distinct features. A commodity like cotton has different grades and the demand for a particular grade cannot be satisfied by supply of another grade. Similarly in case of commodity like paper, buyers may require paper of different widths and sellers may supply paper of different widths. In such cases, the price and quantity need not be the only differentiating factors. These requirements give rise to different types of constraints, as they restrict the assignment of asks and bids. Such constraints where a set of asks can be assigned to only certain sets of bids are called as “assignment constraints”. The assignment constraints can be of several types. In one type only certain subset of asks can be assigned to a set of bids. Though there can be restrictions on assigning an ask to a bid, the supply from several asks can be combined to meet the demand of a bid. There are other cases where it is not allowed to meet the demand of a buyer from multiple sources. Consider a case where buyer requires a contiguous paper roll of specified width and length. In such cases it may not be possible to combine supply from different asks to satisfy the demand of the buyer. Such constraints are called as “indivisible demand” bid constraint. In such “indivisible demand bid constraints”, a single ask is to be assigned to a bid for which demand can not be fulfilled by combining more than one asks. We consider an example from the process industry to illustrate the notion of double auctions with different types of constraints.

3.2 Different Types of Assignment Constraints and Indivisibility Constraints

In order to illustrate the notions of different types of assignment constraints, indivisible demand bid constraints with or without assignment constraints and the present work, consider the following example of sale of paper rolls. Let us assume that there are 5 sellers and 5 buyers. The paper rolls sold by different sellers have different specifications in terms of grade, width, selling price and quantity. Similarly the buyers specify their respective requirements in terms of buying price, quantity, grade and
width. The details of sellers are given in Table 3.1. It can be seen from Table – 3.1 that, first seller (seller – 1) has 3 tons of paper roll of width 2000cm and grade 1. The selling price per ton is 100. The detailed requirements of buyers are shown in Table 2. It can be seen from Table 3.2 that, the first buyer requires 5 tons of paper roll with width of 2000cm. The grade of the required paper is 3 and the buying price is Rupees 175 for each ton. In this example different grades of paper are indicated by positive integers 1,2,3 and 4. The buyers and sellers indicate the grade of paper required by them. The paper of higher grade is represented by higher integer value. The width of the paper roll is specified in the units of centimeters. Both buyers and sellers specify it in their respective bids and asks.

<table>
<thead>
<tr>
<th>Seller</th>
<th>Price</th>
<th>Width (CM)</th>
<th>Supply (Quantity Tons)</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>2000</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>105</td>
<td>2000</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>110</td>
<td>2000</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>114</td>
<td>1000</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>119</td>
<td>1000</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Buyer</th>
<th>Price</th>
<th>Width</th>
<th>Demand</th>
<th>Grade</th>
<th>Higher Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>175</td>
<td>2000</td>
<td>5</td>
<td>3</td>
<td>N</td>
</tr>
<tr>
<td>2</td>
<td>170</td>
<td>1200</td>
<td>5</td>
<td>4</td>
<td>N</td>
</tr>
<tr>
<td>3</td>
<td>165</td>
<td>800</td>
<td>3</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>4</td>
<td>163</td>
<td>800</td>
<td>3</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>5</td>
<td>161</td>
<td>1000</td>
<td>10</td>
<td>1</td>
<td>Y</td>
</tr>
</tbody>
</table>

It can be seen that there are certain assignment constraints in above example, which must be considered while matching. The assignment constraints for matching are as follows:

(1) A bid requiring paper of a particular grade cannot be matched with asks supplying paper of lower grades. In some cases it can be matched with asks supplying paper of
higher grades. It can be seen that the first bid can be matched with asks supplying paper of grade 3 only.

(2) It can be seen from the above example that the bid from the buyer requiring paper of width \( w \) can be assigned with asks supplying paper with width \( w \) or more. It can be seen that the second buyer requiring paper of width 1200 cm can be matched with only those asks, which supply paper of width 1200 cm or more. It can also be seen that the supply from any ask can be used to satisfy the demand of buyers 3 and 4 by cutting the paper into rolls of 800cm.

(3) It can be seen that, in the above example, any bid has to be matched with the set of asks supplying paper of same or higher width. This is an implicit assignment constraint. The above example illustrates a scenario of assignment constraints. Here demand of 10 tons of a bid (say bid 5) can be satisfied by combining supply of 5 tons from ask 4 and 5 tons form ask 5. In such cases it has to be ensured that width, grades and other attributes must match.

In addition to assignment constraints there are scenarios, which give rise to indivisibility constraints. This scenario is illustrated by the following example (Table 3.3). There can be indivisibility constraints, in addition to different types of assignment constraints. Indivisibility constraints mean that, the demand of bid with such constraints cannot be fulfilled by combining supply from different asks. A demand of bid with such constraint can be satisfied by only a single ask. Such a scenario is shown in Table 3.3. One can see that the last column of Table 3.3, gives rise to indivisibility requirements. This table is similar to earlier table (Table 3.2). However the buyer specifies the requirement of contiguous paper roll. This gives rise to indivisibility constraint. It can be seen from Table – 3.3, that the first buyer requires the 5 tons of contiguous paper roll and not the broken one. (In this case grade attribute is omitted for illustration). This gives rise to indivisible demand constraint in the sense that this demand cannot be satisfied by pooling supply of 5 tons from say first two asks i.e. (3 tons from first ask and 2 tons from second ask).
The indivisible demand constraint can arise implicitly due to the width of paper rolls as well as minimum quantity specified by buyer. In case of the first bid, the buyer has specified that he needs 5 paper rolls of width 2000cm. He also needs entire roll as contiguous and not broken one. So this bid cannot be matched with first two asks having total supply of 5. In the same way this bid cannot be matched with ask 4. It has to be satisfied by assigning third ask only. In the same way bid 4 is another indivisible demand bid.

Table. 3.3 - Buyers bids for paper exchange

<table>
<thead>
<tr>
<th>Bids</th>
<th>Buyer</th>
<th>Price</th>
<th>Width (Cm)</th>
<th>Demand</th>
<th>Minimum Contiguous Roll Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>175</td>
<td>2000</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>170</td>
<td>2000</td>
<td>5</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>165</td>
<td>1200</td>
<td>5</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>163</td>
<td>800</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>161</td>
<td>800</td>
<td>5</td>
<td>-</td>
</tr>
</tbody>
</table>

In case of bid 2, its demand of width 2000cm cannot be satisfied by combining two asks, each supplying rolls of width 1000cm. Thus the indivisible demand bid constraint may constrain one or more attributes to be indivisible.

We observe that the indivisibility restriction is imposed only on the supply and demand quantities. But in real life, it is possible to have such constraints on multiple attributes (and not only on quantity). Moreover, the indivisibility restriction may be characteristic of the attribute or of the bid. Thus there can be bids, which may or may not have indivisibility restriction on a particular attribute.

In the last few years, there has been a growing interest in study of auctions due to emergence of Internet based electronic markets. Electronic markets leverage on the information technology, to perform different market functions efficiently and reduce cost of transactions. Large numbers of auctions are carried out over the Internet and apart from traditional auctions; many different types of auction systems are implemented in an electronic environment. An auction is an economic mechanism that
is widely used to sell different commodities such as food grains, flowers, real estates, cars, air tickets etc. Auctions are commonly used when the sellers do not want to decide the price and allow the market to determine it. The market is a very important component of economy and it provides legal and institutional framework for transactions. There are different types of auctions like English, Dutch, Sealed bid [W1996], which are used in various economic transactions in different types of markets. A detailed survey of auction mechanisms can be found in [W1996]. More recently there have been many novel applications of electronic auctions like electronic procurement [EGKL2001], bidding on air ticket etc. An auction-based process, which minimizes the procurement cost using auctions, is proposed in [KL2001]. Unlike the traditional English or Dutch auctions where price is the only attribute, a multi-attribute auction [BKS1999] is based on more than one attribute. In combinatorial auctions [RPH1998] [S2002], the seller wishes to sell a combination of goods and buyers bid on one or more goods. An approximate but tractable strategy proof mechanism for single-good-multi-unit allocation problem is presented in [KPS2003]. It also presents a polynomial time approximation scheme for reverse and forward auctions.

Even though auctions have been widely studied in economic theory, auctions under different types of constraints have not been studied widely. The problem of determining optimal matching with indivisible demand bid constraint is first proposed in [KDL2001]. In the absence of any assignment constraints, the matching problem can be solved using the price discovery mechanism [GD1998] and [KDL2001]. This work [KDL2001] basically investigates computational complexity of clearing markets in continuous call auction in three cases viz, (1) no constraints on assignment (unconstrained assignment), (2) constraints which restrict assignment of an ask to bid (assignment constraints) and (3) indivisible demand bid constraints where supply from different asks cannot be combined to satisfy demand (indivisible demand). An algorithm to find optimal matching of asks and bids, when there are no assignment constraints is presented in [KDL2001]. This algorithm is a multi unit extension of single unit double auction markets studied in [SW1989] and [M1992]. In case of assignment constraints, problem of finding optimal assignment is formulated as a
network flow problem [AMO1993],[KDL2001],[CH1995],[CHM1996]. All bids and asks form the intermediate nodes of this network. The arcs are constructed depending upon the prices and quantities. The arcs between the source node and bids are constructed using quantity of bid and 0 (cost). In the same way, the arcs from asks to sink node are constructed. These arcs are constructed using the difference between the prices as cost and quantity as weight. This problem is then solved as a network flow optimization problem. There are many algorithms available to solve this problem [AMO1993]. The complexity of maximum flow problem i.e. matching of demand and supply is $O((nm + n^2 \log n))$. In this case, n represents number of nodes and m number of edges. Three different formulations are suggested for finding optimum assignment of asks and bids in case of indivisible demand [KDL2001]. These are multiple knapsack problem, bin packing problem or generalized assignment problems [MT1989], [GJ1979] and [S2002]. Apart from this the problem of double auctions with constraints has not been studied widely. Further this work [KDL2001] only partly addresses trade determination problem. It does not address payoff determination problem. Electronic auctions are now used widely for different types of problems and in many such problems there can be different types of constraints. In many cases price and quantity need not be the only deciding attributes. It can be seen that the existing approaches are not adequate to handle different types of problems. It is therefore essential that the problem of double auctions with constraints be studied in detail.

In the present formulation the bids can be of mixed types i.e. some bids can have indivisible demand constraints, whereas other bids need not have these constraints. They can also optionally have other assignment constraints based on different attributes (e.g. width, in this case). We observe that price and quantity are not the only attributes that are to be constrained for indivisibility. The above example shows that even another attribute, say width, can also have such restriction. Thus we address the problem in much more general settings.

In the present work the algorithms for finding optimum assignment of asks and bids, under different types of constraints, are developed. Later we develop algorithms for computing VCG Payments for different buyers and sellers.
3.3 Problem Formulation

In this section the problem of optimal matching for continuous call auctions with indivisible bid constraint is formulated. It is observed that our formulation is closer to the real life applications. Further, we develop some new results that help in devising an efficient algorithm.

The object under sale in an auction is referred to as an item. The attributes describe different characteristics of the items. There can be number of attributes to describe each item. However, we identify three special attributes for each item namely, the price, quantity and size or width. Each attribute assumes value from a set of specified domains. Let there be m asks and n bids. Let AS be the set of all asks and BD be the set of all bids. The set of asks AS and bids BD can be decomposed to AS$_{o1}$, AS$_{o2}$, ..., AS$_{on}$ and BD$_{o1}$, BD$_{o2}$, ..., BD$_{om}$. So that we can write AS and BD as

\[ \text{AS} = \text{AS}_{o1} \cup \text{AS}_{o2} \cup \ldots \cup \text{AS}_{on} \quad \text{and} \quad \text{B} = \text{BD}_{o1} \cup \text{BD}_{o2} \cup \ldots \cup \text{BD}_{om}. \]

The sets BD$_{oi}$ are disjoint i.e. BD$_{oi} \cap$ BD$_{oj} = \emptyset$ for all i $\neq$ j.

The set of asks AS$_{oi}$ are not required to be mutually exclusive. This decomposition can be done based on values of set of attributes.

**Definition 3.1 (Ask)**

An ask $A_i$ is an ordered list of k attribute values, $A_i = (v_{i1}, v_{i2}, \ldots, v_{ik})$, where $v_{ij}$ is the value of the $j^{th}$ attribute of $i^{th}$ ask.

**Definition 3.2 (Bid)**

A bid $B_i$ is an ordered list of k attribute values, $B_i = (v_{i1}, v_{i2}, \ldots, v_{ik})$, where $v_{ij}$ is the value of the $j^{th}$ attribute of $i^{th}$ bid.

A seller describes the details of an item he wants to sell in an ask whereas the buyer describes the details of items and the price that he is willing to pay in a bid. Let $bq_i$ and $aq_j$ be the quantities of bid $B_i$ and ask $A_j$, respectively. Also, let $bp_i$ be the price of bid $B_i$ and $ap_j$ be the price of ask $A_j$. Let $as_j$ be the size of the ask $A_j$ and $bs_i$ be the size of bid $B_i$ specified in same units.

Once the asks and the bids are grouped in this way, the matching can be done for each subset of asks and bids separately, considering each group as an unconstrained
matching. This approach can be efficient when the constraints are of equality type. The difficulty with this approach is that, in some cases it may not be possible to exactly group asks and bids. For example, in Table 3.1, bids 3 and 4 require paper roll of 800cm. It can be possibly matched with any ask where width is 800 cm or more. There can also be another potential problem. Suppose that a bid, which requires paper of width 800 cm (quantity 5 tons) is matched with an ask of 1200 cm (quantity 5 tons). Then remaining part of ask i.e. 400cm of width (quantity 5 tons) will either be wasted or should be moved to another set. In our approach we do not follow this procedure of multiple order books due to difficulties mentioned above and treat it as a single order book. We also take into account other attributes like width while matching.

Let us define variable \( x_{ij} \) for \( 1 \leq i \leq n \), and \( 1 \leq j \leq m \) as follows.

\[
x_{ij} = \begin{cases} 
1, & \text{if the \( j \)th ask is assigned to \( i \)th bid} \\
0, & \text{otherwise}
\end{cases}
\]

The quantity \( q_{ij} \) of item that is matched between \( j \)th ask and \( i \)th bid is given by

\[
0 \leq q_{ij} \leq \min(bq_i, aq_j) \text{ for all } j (1 \leq i \leq n, 1 \leq j \leq m).
\]

In case \( q_{ij} = bq_i \), it means that the demand of bid \( B_i \) is completely satisfied. In case \( q_{ij} = aq_j \), it means that supply from the ask \( A_j \) is completely fulfilled. If it is neither, it signifies partial fulfillment of bids and asks.

**Definition 3.3 (Price Spread)**

The price spread between bid \( B_i \) and ask \( A_j \) is

\[
p_{ij} = (bp_i - ap_j).
\]

**Definition 3.4 (Surplus and Total Surplus)**

The surplus from the matching quantity \( q_{ij} \) between \( i \)th bid and \( j \)th ask is \( p_{ij}q_{ij}x_{ij} \).

The total surplus \( R \) of matching process is given by

\[
R = \sum_i \sum_j p_{ij}q_{ij}x_{ij} \tag{3.1}
\]

While matching asks and bids, an attempt is made to maximize the total surplus (i.e. to obtain the matching such that the total of difference between ask price and bid price is maximum).
Let us consider a situation where it is necessary to account for even the wastage during the matching process. The wastage could arise due to difference in sizes (width in example) of asks and bids. An ask can be matched with a bid only if its size is either same as the bid size or exceeds it. Indivisible demand bid constraints arise because asks of two different sizes cannot be combined. It can be seen from Table 3.1, if ask 4 is matched with bid 4 and ask 5 is matched with bid 3, paper roll of width 200cm and of quantity 6 tons is wasted. On the other hand if the first ask is matched with bid 2, then there will not be any wastage.

**Definition 3.5 (Wastage and Wastage Penalty):**

Suppose that \(i^{th}\) bid with value of width attribute \(b_{si}\) is assigned to \(j^{th}\) ask with value of width attribute \(a_{sj}\). Then quantity \((a_{sj} - b_{si})\) (difference between respective values of width attributes) is the wastage resulting from this assignment. The wastage is minimum when \(a_{sj} = b_{si}\) and the minimum value is 0. Its values are always nonnegative.

Let \(w_{ij}\) be the per unit wastage penalty expressed in monetary units (i.e. in the same units as price) for wastage resulting from assignment of \(i^{th}\) bid and \(j^{th}\) ask.

This per unit wastage penalty is a linear function of wastage \((a_{sj} - b_{si})\). So we have

\[ w_{ij} = wp(a_{sj} - b_{si}), \]

which is a linear function of wastage. Its important properties are as follows.

1. It assumes only nonnegative values. It is an one to one function; meaning that for any quantity of wastage there is unique associated wastage penalty value and vice versa.
2. Its value is 0 when wastage is \((a_{sj} - b_{si}) = 0\). It is the minimum value for wastage penalty. Its value is minimum, when wastage is minimum.
3. Its value increases as value of \((a_{sj} - b_{si})\) increases.
4. If wastage from two different assignments is same i.e. \((a_{s1} - b_{s1}) = (a_{s3} - b_{s4})\), then the wastage penalty is same i.e. \(w_{i2i1} = w_{i4i3}\).
(5) Let $B_{i1}$ and $B_{i2}$ be two bids and asks $A_{i3}$ and $A_{i4}$ are assigned to them. Then total per unit wastage penalty, resulting these two assignments is sum of per unit wastage penalties of individual assignments. In other words
\[ \text{wp}(as_{i3} - bs_{i1} + as_{i4} - bs_{i2}) = \text{wp}(as_{i3} - bs_{i1}) + \text{wp}(as_{i4} - bs_{i2}). \]

The wastage penalty from assignment of quantity $q_{ij}$ of $i^{th}$ bid and $j^{th}$ ask is defined is $w_{ij}q_{ij}$. The total wastage penalty $W$ from matching is given by
\[ W = \sum_i \sum_j w_{ij}q_{ij}x_{ij} \]

Our objective from matching is to maximize the surplus and minimize the wastage penalty.

So the problem can be formulated as a nonlinear integer programming problem as follows.
\[
\begin{align*}
\text{Max} & \quad \sum_i \sum_j (p_{ij} - w_{ij}) q_{ij}x_{ij} \quad (3.2) \\
\sum_j q_{ij}x_{ij} & = b_{qi} \quad \text{for every indivisible demand bid } B_i \quad (3.3) \\
\sum_j x_{ij} & = 1 \quad \text{for every indivisible demand bid } B_i \quad (3.4) \\
\sum_j q_{ij}x_{ij} & \leq b_{qi} \quad \text{for all other bids} \quad (3.5) \\
\sum_j q_{ij}x_{ij} & \leq a_{qj} \quad \text{for all asks } A_j \quad (3.6) \\
p_{ij}, w_{ij} & \geq 0 \quad \text{for all } i \text{ and } j \quad (3.7)
\end{align*}
\]

Let $A$ be the set of bids and asks which cannot be assigned to each other.
\[
\begin{align*}
x_{ij} & = 0 \quad \text{if } (i, j) \in A \\
& = 0, 1 \text{ otherwise}
\end{align*}
\]

This set $A$ includes all equality as well as inequality constraints on different attributes of asks and bids. If no bid has indivisibility constraints, then constraints in (3.4) can be omitted. The optimization problem Max $R$ (as defined in 3.1) subject to constraints in (3.3) - (3.7) is a particular case of this formulation, where $w_{ij} = 0$ for $1 \leq i \leq m, 1 \leq j \leq n$. It is an optimization problem Max $R$ subject to constraints (3.3) - (3.7). This will be referred as formulation (1). It is a generic surplus maximization problem.

Thus the above formulation is a more general one and its special cases are no assignment constraints, assignment constraints with or without indivisible demand bid constraints. The following result is the basis of our proposed algorithms.
**Definition 3.6 (Contribution)**

Let quantity $q$ of ask $A_k$ with price $a_{p_k}$ be assigned to bid $B_i$. Let $b_{p_i}$ be the price attribute of this bid. Let $a_{s_k}$ and $b_{s_i}$ be the size attributes of ask $A_k$ and bid $B_i$ respectively. Then contribution $c_{ov_{ik}}(q)$ to the value of objective function from this assignment is defined as

$$c_{ov_{ik}}(q) = (b_{p_i} - a_{p_k})q - w_{ik}q = c_{ovs_{ik}}(q) - c_{ov_{ik}}(q)$$  \( (3.8) \)

$q \leq \min(a_{q_j}, b_{q_i})$  \( (3.9) \)

If $q = 1$, we call it as unit contribution and indicate it by $u_{cov_{ik}}$. The maximum possible unit contribution for any bid can be obtained, when it is assigned to an ask with minimum price and having value of size attribute same as that of bid size, if such an ask exists. It can be noted that both the terms in (3.8) are always nonnegative. The contribution decreases if ask size increases and the contribution increases if ask size decreases. This happens because wastage penalty decreases if ask size decreases. The contribution to the value of objective function consists of two components - contribution to the surplus and wastage penalty. The contribution to surplus is

$$c_{ovs_{ik}}(q) = (b_{p_i} - a_{p_k})q$$

and contribution to wastage penalty is $c_{ovo_{ik}}(q) = w_{ki} = w_{p}(a_{s_k} - b_{s_i})q$.

Let us indicate the assignment of quantity $q$ of $j^{th}$ ask to $i^{th}$ bid ($i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$) by $A_{sg}$ as $A_{sg} = \{(B_{i}, A_{j}, q)\}$.

The quantity $q$ satisfies the condition $q \leq \min(b_{q_i}, a_{q_j})$. Further

(i) If $q = b_{q_i}$, it is a complete assignment of bid.

(ii) If $q = a_{q_j}$, it is a complete assignment of ask.

(iii) If $q = b_{q_i} = a_{q_j}$, it is a full assignment.

(iv) In all other cases it is a partial assignment.

In the same way, assignment of more than one asks i.e. $A_1$ with quantity $q_1$, $A_2$, with quantity $q_2$, ..., $A_{k_1}$ with quantity $q_{k_1}$ for a bid $B_1$ is indicated as

$A_{sg} = \{(B_{1}, A_{j}, q_{j}), \ j = 1, 2, \ldots, k_1\}$. In a similar manner assignment of one ask to many bids can be expressed.
Theorem 1 - Suppose that in optimum assignment of the problem defined in (3.2), the assignment of asks in respect of bids $B_i$ and $B_j$ satisfies the following conditions.

1. $A_{sg1} = \{(B_i, A_{k1}, q)\}$ - quantity $q$ of ask $A_{k1}$ is assigned to bid $B_i$
2. $A_{sg2} = \{(B_j, A_{il}, q_{il})\}, 1 \leq l \leq k, l$ is positive integer} - quantities $q_{il}$ of ask $a_{il}, (1 \leq l \leq k, l$ is positive integer) respectively, are assigned to bid $B_j$
3. The quantity $q$ satisfies the relation $q = \sum_{l=1}^{k} q_{il}$ and conditions $q \leq \min (bq_i, aq_{k1})$.

Then the value of objective function remains unchanged if $A_{sg1}$ and $A_{sg2}$ are changed as follows.

1. $A_{sg1} = \{(B_i, A_{il}, q_{il})\}, 1 \leq l \leq k, l$ is positive integer
2. $A_{sg2} = \{(B_j, A_{k1}, q)\}$
3. There is no change in remaining assignment.

The proof: The contribution to value of objective function for assigning quantity $q$ of $A_{k1}$ to $B_i$ is given by,

$$cov_{ik1}(q) = (bp_i - ap_{k1})q - w_{ik1}q = cov_{ik1}(q) - cov_{ik1}(q),$$

where $cov_{ik1}(q) = (bp_i - ap_{k1})q$ and $w_{ik1}q = cov_{ik1}(q) = wp(as_{k1} - bs_i)q$

The total contribution to value of objective function $tcov_{jk}(q)$, for assigning $A_{il}$ with quantity $q_{il}, (1 \leq l \leq k)$ to $B_j$ is given by,

$$tcov_{jk}(q) = \sum_{l=1}^{k} ((bp_j - ap_{il})q_{il} - w_{jil}q_{il}) (1 \leq l \leq k)$$

$$= \sum_{l=1}^{k} (w_{jil}q_{il} - wcov_{il}(q)), \text{ where}$$

$$cov_{il}(q) + wcov_{il}(q) = wp(as_{il} - bs_j)$$

We can write $q = \sum_{l=1}^{k} q_{il}$

$$= (bp_i - ap_{k1})q - w_{ik1}q + \sum_{l=1}^{k} ((bp_j - ap_{il})q_{il} - w_{jil}q_{il}) (1 \leq l \leq k)$$

$$= \sum_{l=1}^{k} (bp_i - ap_{k1}) q_{il} - w_{ik1} \sum_{l=1}^{k} q_{il} + \sum_{l=1}^{k} ((bp_j - ap_{il}) q_{il} - w_{jil}q_{il})$$

$$= \sum_{l=1}^{k} ((bp_i - ap_{k1}) - w_{ik1} + (bp_j - ap_{il}) - w_{jil}) q_{il}$$
\[ \sum_{i=1}^{k} ((b_{pi} - a_{p_{k1}}) + (b_{pj} - a_{p_{i}}) - (w_{ik1} + w_{j{i}}) \) q_{il} \]  (3.10).

We can write \( w_{ik1} = w_{pi}(a_{k1} - b_{s_i}) \) and \( w_{ji} = w_{pj}(a_{s_i} - b_{s_j}) \).

The wastage resulting from assignment of quantity \( q \) of ask \( A_{k1} \) to bid \( B_i \) is \((a_{s_k} - b_{s_i})q \). It can be written as \( \sum_{i=1}^{k} (a_{s_k} - b_{s_i})q_{il} \).

The total wastage from assigning \( A_{il} \) with quantity \( q_{il} \) (1 \( \leq \) \( l \) \( \leq \) \( k \)) to \( B_j \) is given by,

\[ \sum_{i=1}^{k} (a_{s_i} - b_{s_j})q_{il}. \]

The total wastage resulting from two assignments (assignment of ask \( A_{k1} \) to bid \( B_i \) and set of asks \( A_{il} \) (1 \( \leq \) \( l \) \( \leq \) \( k \)) to bid \( B_j \)) is

\[ \sum_{i=1}^{k} (a_{s_k} - b_{s_i})q_{il} + \sum_{i=1}^{k} (a_{s_i} - b_{s_j})q_{il}. \]

It can be rewritten as

\[ \sum_{i=1}^{k} (a_{s_i} - b_{s_j})q_{il} + (a_{s_k} - b_{s_j})q = \text{Total wastage resulting from assignment of set asks } A_{il} \text{ (1} \leq \text{ } l \leq \text{ } k) \text{ to } B_i \text{ and ask } A_{k1} \text{ to bid } B_j. \]

Due to this total wastage penalty remains same by property (4). So we can write

Total wastage penalty from (assignment of quantity \( q \) of \( A_{k1} \) to bid \( B_i \) and quantity \( q_{il} \), (1 \( \leq \) \( l \) \( \leq \) \( k \)) of asks \( A_{il} \) to bid \( B_j \) = Total wastage penalty from (assignment of quantity \( q \) of \( A_{k1} \) to bid \( B_j \) and quantity \( q_{il} \), (1 \( \leq \) \( l \) \( \leq \) \( k \)) of asks \( A_{il} \) to bid \( B_i \).  (3.11).

Using property (5) two sides of identity (3.11) can be written as

\[ w_{pi}(a_{k1} - b_{s_i})q + \sum_{i=1}^{k} w_{pi}(a_{s_i} - b_{s_j})q_{il} = \sum_{i=1}^{k} w_{pi}(a_{s_i} - b_{s_j})q_{il} + w_{pj}(a_{s_i} - b_{s_j})q. \]

This can be written as

\[ w_{ik1}q + \sum_{i=1}^{k} w_{ji}q_{il} = \sum_{i=1}^{k} w_{ii}q_{il} + w_{j1}q. \]

Using \( q = \sum q_{il} \), it can be further rewritten as

\[ \sum_{i=1}^{k} (w_{ik1} + w_{ji})q_{il} = \sum_{i=1}^{k} (w_{ii} + w_{jk1})q_{il}. \]  (3.12).

Substituting this in (3.10) we get

\[ \sum_{i=1}^{k} ((b_{pi} - a_{p_{k1}}) + (b_{pj} - a_{p_{i}}) - (w_{ii} + w_{jk1})q_{il}) \]

\[ = \sum_{i=1}^{k} ((b_{pi} - a_{p_{i}}) - w_{ii}q_{il} + \sum_{i=1}^{k} ((b_{pj} - a_{p_{k1}}) - w_{jk1})q_{il} \]
\[
\sum_{i=q} \left( (b_i - a_{i}) - w_{ii} q_{il} + (b_{j} - a_{k1}) - w_{jk1} \right) q_{il} = t_{cov_i}(q) + c_{ovj_{k1}}(q)
\]

\[
(t_{cov_i}(q) - c_{ovj_{k1}}(q)) = t_{covs_i}(q) - t_{covo_i}(q) + c_{ovs_{jk1}}(q) - c_{ovj_{k1}}(q)
\]

t_{cov_i}(q) is the total contribution to the value of objective function from the assignment of quantity q of set of asks \(a_{i1}, a_{i2}, \ldots, a_{ik}\) to \(B_i\) and \(c_{ovj_{k1}}(q)\) is the contribution to the value of objective function from the assignment of quantity q of ask \(A_{k1}\) to bid \(B_j\).

This shows that an interchange of quantity q of an ask, assigned to one bid, with set of asks assigned to another bid, will not affect the value of objective function. We call this as an interchange result.

In other words, the value of objective function will not be affected by interchanging quantity q of ask \(A_{k1}\) (assigned to bid \(B_i\)) with quantity q of set of asks \(a_{il}\) (1 \(\leq l \leq k\), (assigned to bid \(B_j\). (In the new assignment quantity q of ask \(A_{k1}\) is assigned to bid \(B_j\) and set of asks \(A_{il}\) with quantity q \((l = 1, 2, \ldots, k)\) is assigned to bid \(B_i\).) In this case it is not assumed that \(A_{sg1}\) or \(A_{sg2}\) is a complete assignment in respect of bids \(B_i\) and \(B_j\). In case it is a partial assignment of bids \(B_i\) and \(B_j\), then the remaining assignment is not changed by this interchange. In this case, if q = \(a_{q_{k1}}\), then it is a complete interchange of ask \(A_{k1}\). It can be easily proved, that the above result is true even if size attribute or wastage factor is omitted (i.e. only for surplus). This result shows that, if we have an optimum matching of asks and bids which maximizes the net surplus, it is possible to obtain an assignment that can minimize the wastage (as in this example), without affecting the total surplus. This shows that it is possible to consider other attributes while matching. The formulation in (3.2) helps in obtaining such matching assignment in one stage, rather than attempting it as two stage optimization problem. Further, it can be seen from this result that the surplus from the assignment does not change because of interchange. It can be seen that our objective function is of type \(O = (A - B)\), where both A and B have non negative values. So O will have maximum value when A is maximum and B is minimum. Due to this, once we get optimum value of objective function defined in (3.2), subject to constraints in (3.3) to (3.7), we will also get
optimum of (3.1), subject to constraints (3.3) to (3.7). Because of this reason, objective function is defined as in (3.2). However our formulation and algorithm can be applied even in case of two stage optimization (first maximizing surplus and then minimizing wastage) process. The second attribute used here is width, as seen in the example of paper roll. However, any numeric attribute can be used in its place. We now state few particular cases of the above result.

**Theorem 2**: Suppose that there is full assignment of quantity $q$ in optimum matching as follows.

1. $A_{sg1} = \{(B_i, A_k, q)\}$
2. $A_{sg2} = \{(B_j, A_h, q)\}$

Then the value of objective function remains unchanged if $A_{sg1}$ and $A_{sg2}$ are changed, without changing any other assignment of asks and bids, as

1. $A_{sg1} = \{(B_i, A_h, q)\}$
2. $A_{sg2} = \{(B_j, A_k, q)\}$

**Proof**: The contribution to value of objective function for assigning $A_k$ to $B_i$ is given by,

$$\text{cov}_{ik}(q) = (b_{pi} - a_{pk}) q - w_{ik} q$$

Similarly, the contribution to value of objective function for assigning $A_h$ to $B_j$ is given by,

$$\text{cov}_{jh}(q) = (b_{pj} - a_{ph}) q - w_{jh} q$$

In this case $b_{qi} = b_{qj} = a_{qk} = a_{qh} = q$

The total contribution is

$$\text{cov}_{ik}(q) + \text{cov}_{jh}(q)$$

$$= (b_{pi} - a_{pk}) q - w_{ik} q + (b_{pj} - a_{ph}) q - w_{jh} q.$$  

Following a similar argument as in theorem (1) it can be shown that

$$w_{ik} q + w_{jh} q = w_{ih} q + w_{jk} q.$$  

This expression can be simplified as

$$= ((b_{pi} - a_{ph}) q - w_{ih} q + (b_{pj} - a_{pk}) q - w_{jk} q)$$

$$= \text{cov}_{ih}(q) + \text{cov}_{jk}(q)$$
where \( \text{cov}_{ih}(q) \) and \( \text{cov}_{jk}(q) \) are contributions to the value of objective function for assigning \( A_h \) to \( B_i \) and \( A_k \) to \( B_j \), respectively. This shows that a complete interchange of an ask, assigned to one bid to another, will not affect the value of objective function.

It can be easily shown that the value of objective function is not affected, if assignment of asks and bids are partly interchanged. It can also be seen that the above result holds good even if interchange is done partly assuming that it is allowed. The partial interchange of quantity \( q \) is not allowed for bids with indivisible demand bid constraints.

**Theorem 3** – Suppose that in optimum matching assignment of asks in respect of bids \( B_i \) and \( B_j \) is as follows.

(1) \( A_{sg1} = \{ (B_i, A_k, aq_k) \} \)

(2) \( A_{sg2} = \{ (B_j, A_h, aq_h) \} \)

Then the value of objective function will remain unchanged if \( A_{sg1} \) and \( A_{sg2} \) are changed, without changing any other assignment of asks and bids, as

(1) \( A_{sg1} = \{ (B_i, A_k, aq_k-q), (B_i, A_h, q) \} \)

(2) \( A_{sg2} = \{ (B_j, A_h, aq_h-q), (B_j, A_k, q) \} \)

**Proof** : Let us assume that the asks \( A_k \) and \( A_h \) are assigned to bids \( B_i \) and \( B_j \) with quantities \( aq_k \) and \( aq_h \) respectively. In the new assignment quantity \( q \) of ask \( A_k \) (out of \( aq_k \)) is assigned to bid \( B_j \) and quantity \( q \) of ask \( A_h \) (out of \( aq_h \)) is assigned to bid \( B_i \). There is no change in other assignments. In this case the contribution to the value of objective function for assigning quantity \( aq_k \) of \( A_k \) to \( B_i \) is given by

\[
\text{cov}_{ik}(aq_k) = (bp_i - ap_k)aq_k - w_{ik}aq_k
\]

Then the contribution to value of objective function for assigning quantity \( aq_h \) of \( A_h \) to \( B_j \) is given by,

\[
\text{cov}_{jh}(aq_h) = (bp_j - ap_h)aq_h - w_{jh}aq_h
\]

The total contribution is

\[
\text{cov}_{ik}(aq_k) + \text{cov}_{jh}(aq_h)
\]

\[
= (bp_i - ap_k)aq_k - w_{ik}aq_k + (bp_j - ap_h)aq_h - w_{jh}aq_h
\]

substitute \( aq_k = (aq_k - q) + q \) and \( aq_h = (aq_h - q) + q \), in above expression
Rearranging and simplifying as in theorem 1, we get

\[(bp_i - ap_k) - w_{ik} \cdot (aq_k - q) + q\]  
\[(bp_j - ap_h) - w_{jh} \cdot (aq_h - q) + q\]

\[= (bp_i - ap_k - w_{ik} \cdot (aq_k - q) + (bp_j - ap_h - w_{jh} \cdot (aq_h - q) + q]

\[= ((bp_i - ap_k - w_{ik} \cdot (aq_k - q) + ((bp_j - ap_h - w_{jh} \cdot (aq_h - q) + (bp_j - ap_h - w_{jh} \cdot (aq_h - q) + q)]

where \(cov_{ik}(aq_k - q)\) and \(cov_{ih}(q)\) are the contributions, to the value of objective function, by assigning quantities \((aq_k - q)\) of ask \(A_k\) and \(q\) of ask \(A_h\) to bid \(B_i\) respectively. Similarly \(cov_{jh}(aq_h - q)\) and \(cov_{jk}(q)\) are the contributions, to the value of objective function, by assigning quantities \((aq_h - q)\) of ask \(A_h\) and \(q\) of ask \(A_k\) to bid \(B_j\) respectively. This shows that partial interchange of an ask, assigned to one bid to another, does not affect the value of objective function. We now consider the interchange of any quantity \(q\), in case of two bids.

**Theorem 4:** Suppose that we have an assignment \(A_{sg} = \{(B_j, A_i, q), i = 1, 2, 3, j = 1, 2, 3\}\), in optimum matching in respect of bids \(B_1, B_2\) and \(B_3\). Suppose that \(A_{sg}\) is changed as

\(A_{sg} = \{(B_1, A_2, q), (B_2, A_3, q), (B_3, A_1, q)\}\) without changing any other assignment, then the value of objective function remains unchanged.

**Proof:** This result immediately follows from theorem 1. The interchange is equivalent to the following.

1. Interchange quantity \(q\) of ask \(A_1\) from bid \(B_1\) to bid \(B_2\) and quantity \(q\) of ask \(A_2\) from bid \(B_2\) to bid \(B_1\). This interchange does not affect the value of objective function (from theorem 1).
2. Interchange quantity \(q\) of ask \(A_1\) from bid \(B_2\) to bid \(B_3\) and quantity \(q\) of ask \(A_3\) from bid \(B_3\) to bid \(B_2\). This interchange does not affect the value of objective function (from theorem 1).
Theorem – 5 - Suppose that in optimum matching set of k and m asks are assigned to
bids $B_i$ and $B_j$ respectively, as follows.

1. $A_{sg1} = \{B_i, A_{ic}, q_c \} c=1, 2, 3, \ldots, k\}.$
2. $A_{sg2} = \{(B_j, A_{jh}, q_h) h=1, 2, 3, \ldots, m\}.$
3. The quantities assigned satisfy the relation $q = \sum_{c=1}^{k} q_c = \sum_{h=1}^{m} q_h.$

Then the value of objective function does not change if $A_{sg1}$ and $A_{sg2}$ are changed as
follows without changing any other assignment.

(a) $A_{sg1} = \{(B_i, A_{jh}, q_h) h=1, 2, 3, \ldots, m\}$
(b) $A_{sg2} = \{(B_j, A_{ic}, q_c) c=1, 2, 3, \ldots, k\}.$

Proof: It may be noted that the above interchange is equivalent to following set of
successive interchanges, each of which does not change the value of objective function.

If $q_{i1} < q_{j1},$ then the above interchange is equivalent to the following.

1. Interchange ask $A_{i1}$ of quantity $q_{i1}$ with ask $A_{j1}.$ Then interchange quantity $\left( q_{i1} - q_{j1} \right),$ between asks $A_{j1}$ and $A_{i2}.$ These interchanges do not change the value of objective
functions. If $\left( q_{i1} - q_{j1} \right) > q_{i2}$ then continue the assignment, till all quantity of ask $A_{j1}$ is
completely assigned. If $\left( q_{i1} - q_{j1} \right) < q_{i2},$ then interchange quantity $\left( q_{i1} - q_{j1} \right)$ between
ask $A_{j1}$ and ask $A_{i2}.$ The next interchange is between ask $A_{i2}$ with remaining quantity
and ask $A_{j2}.$ This procedure can be continued till interchange of quantity $q$ is
completed. At each stage equal quantity is interchanged. These successive interchanges
do not change the value of objective function.

It can be seen that the interchange of asks as in (a) and (b) is equivalent to set of
successive interchanges. At each stage there is no change in the value of objective
function. Hence by the interchange in (a) and (b), the value of objective function
remains unchanged.

It can be easily seen that the above result holds good for $k = m = 2.$ Let us assume that
the quantities $q_1$ and $q_2$ of asks $A_{i1}$ and $A_{i2}$ respectively, are assigned to bid $B_i.$ Further
suppose that the quantities $q_3$ and $q_4$ of asks $A_{i3}$ and $A_{i4}$ respectively, are assigned to bid
$B_j.$ The quantities assigned satisfy the following relation

$q = q_1 + q_2 = q_3 + q_4.$
Then interchanging of asks $A_{i1}, A_{i2}, A_{i3}, A_{i4}$ by assigning quantities $q_1$ and $q_2$, currently assigned to bid $B_i$ to bid $B_j$ and assigning quantities $q_3$ and $q_4$ of asks $A_{i3}$ and $A_{i4}$, currently assigned to bid $B_j$ to bid $B_i$, will not change the value of the optimal matching. In this case

$$q_1 \leq aq_{i1}, \quad q_2 \leq aq_{i2} \text{ and } (q_1 + q_2) \leq bq_j$$

where $bq_j$ is the quantity of bid $B_j$. If equality condition is satisfied then it is complete interchange of asks. The same relationship also holds good for $q_3$ and $q_4$.

Combining asks to fulfill demand – Obtaining Maximum Improvement: In this section, we determine how maximum improvement in the value of objective function as defined in (3.1) can be obtained for a single bid $B_i$. Let $a_{p1}$, $aq_1$ and $as_1$ be the price, quantity and size attributes of ask $A_1$ respectively. Let $ucov_{i1}$ be the contribution to the value of objective function by assignment of quantity 1 (unit quantity) of this ask to bid $B_i$. It is

$$ucov_{i1} = (bp_i - ap_1) - w_{i1}.$$  

Suppose that $ucov_{i1}$ is the maximum value.

Let $ucov_{ij}$ be the contribution to the value of objective function by assignment of unit quantity of ask $A_j$ to bid $B_i$. Then

$$ucov_{ij} = (bp_i - ap_j) - w_{ij}.$$  

Then we have

$$ucov_{i1} > ucov_{ij} \text{ for } j = 2, 3, \ldots, m.$$  

Let $q_1$ be the maximum quantity at price $ap_1$ and size attribute $as_1$. Let $cov_{i1}$ be the contribution from this assignment. Then

$$cov_{i1} = (q_1) \times (ucov_{i1}) \quad (3.13)$$  

This is the maximum improvement. It can be seen that this term is a product of quantity assigned and contribution per unit. This can be improved if and only if, the term $ucov_{i1}$ (i.e. unit contribution) is improved or $q_1$ (assigned quantity) is improved. In this case neither is possible – as there is no ask with higher unit contribution and the quantity assigned is the maximum available. Since neither is possible, (3.13) represents, the
maximum improvement from a single assignment. (If $q_1 = b_q$, then it is the maximum possible improvement from assignment to bid $B_i$).

Suppose that $A_2$ is another ask with ask price $a_p_2$ and size $a_s_2$ having highest contribution to value of objective function by assignment of unit quantity of ask $A_2$ except ask $A_1$. Let $ucov_{i2}$ be the contribution to the value of objective function by assignment of unit quantity of asks $A_2$ to bid $B_i$, then

$$ucov_{i2} = (bp_i - ap_2) - w_{i2}$$

(3.14).

Also we have

$$ucov_{i2} > ucov_{ij} \text{ for } j = 3, \ldots, n \text{ and } ucov_{i2} < ucov_{i1}$$

(3.15).

Let $q_2$ be the maximum quantity that can be assigned at price $a_p_2$ and size $a_s_2$. In that case these two assignments represent the maximum improvement that can be obtained by assignment of quantity $q_1 + q_2$ and the maximum contribution to the value of objective function which is

$$cov_{i1} + cov_{i2} = (q_1) \times (ucov_{i1}) + (q_2) \times (ucov_{i2})$$

(3.16).

It can be seen that the contribution can be improved if any of the two terms on right hand side can be improved. As already seen the first term cannot be improved. The same argument holds good for the second term, as $A_1$ is the only ask which can improve unit contribution of ask $A_2$, and it is already assigned. So no other ask can improve unit contribution. The quantity assigned is also the maximum possible value. The second term can not be improved. Therefore it is the highest possible improvement at this stage.

**Theorem 6**: Let $AS_k = \{A_1, A_2, \ldots, A_{k-1}\}$ be the set of asks which can be assigned to bid $B_i$. Suppose asks in $AS_k$ satisfy the following conditions.

1. For any ask $A_j \in AS_k$ $ucov_{ij} > ucov_{i(j+1)}$, $j = 1, 2, \ldots, k-2$. Let $q_1, q_2, \ldots, q_{k-1}$ be the quantities assigned respectively.

2. For any ask $A_j \notin AS_k$ $ucov_{ij} < ucov_{il}$, where $l = 1, 2, \ldots, k-1$ (i.e. set of asks belonging to $AS_k$)
3. Let \( A_k \) be an ask such that \( \text{ucov}_{ik} > \text{ucov}_{ij} \), where \( j = k+1, \ldots, m \) (set of asks not in \( \text{AS}_k \)). The assignment of quantity \( q_k \) of this ask completely satisfies the demand of bid \( B_i \).

Then we can obtain maximum improvement by assigning all asks in \( \text{AS}_k \), and \( A_k \) to bid \( B_i \). This is the maximum possible improvement by assigning set of asks or an ask to bid \( B_i \).

**Proof:** This result can be easily shown by induction. As already shown the result is true for \( k =1 \) and \( k = 2 \). We assume that the result is true for \((k-1)\) and show that it holds for \( k \).

The total contribution \( \text{tcov}_{ik} \) to the value of objective function after assignments of \( k \) asks is

\[
\text{tcov}_{ik} = \sum_{l=1}^{k-1} (q_l) \times (\text{ucov}_{il}) + (q_k) \times (\text{ucov}_{ik})
\]  

(3.17).

The contribution can be improved, if and only if the second term can be improved. Since first term cannot be improved (by assumption), only second term can be improved. In this case, it is not possible to improve second term, since its unit contribution \( \text{ucov}_{ik} \) is the highest among the remaining asks (and hence cannot be improved). In the same way \( q_k \) is the maximum quantity that can be assigned. So the second term cannot be improved. Therefore \( \text{tcov}_{ik} \) is the maximum improvement in the value of objective function by assignment to a bid \( B_i \). So the result follows by induction.

In case bid \( B_i \) has indivisible demand bid constraint, then only a single ask can be assigned to it. In this case, the ask with highest unit contribution and satisfying demand of bid \( B_i \) completely, gives the highest improvement.

Application of theorem 6 to a bid with indivisible demand constraint: Let \( B_i \) be the bid with demand \( b_{q_i} \) and price \( b_{p_i} \). The net surplus generated from matching is maximum when the bid is matched with the ask having minimum price, which can satisfy demand completely, if such an ask exists. This ask generates maximum net surplus for a bid with indivisible demand. Let \( a_{p_{min}} \) be the price of such ask. The maximum net surplus from this assignment is
Suppose that there is another ask $A_j$ with price $a_{p_j}$, higher than $a_{p_{\text{min}}}$, which satisfies the demand of bid completely. The surplus from this assignment will be

$$R_{b_s} = (p_{b_i} - a_{p_j}) q_{b_i}$$

Also $a_{p_{\text{min}}}$ being the minimum price we have inequality

$$a_{p_{\text{min}}} < a_{p_i}$$

So $(p_{b_i} - a_{p_{\text{min}}}) > (p_{b_i} - a_{p_j})$

Since $q_{b_i}$ are positive quantities for all $i$

$$(p_{b_i} - a_{p_{\text{min}}}) q_{b_i} > (p_{b_i} - a_{p_j}) q_{b_i}.$$ 

So $R_{b_{\text{imax}}} > R_{b_s}$

Suppose that there is another ask $A_j$ with price $a_{p_j}$ having same price as $a_{p_{\text{min}}}$, however its supply (i.e. quantity) does not satisfy the demand of bid completely. Let $a_{q_j}$ be its quantity. This does not satisfy the demand fully i.e. $a_{q_j} < b_{q_i}$. The surplus from this assignment is

$$R_{b_s} = (p_{b_i} - a_{p_j}) a_{q_j},$$

since $a_{p_j} = a_{p_{\text{min}}}$ it can be written as

$$= (p_{b_i} - a_{p_j}) a_{q_j}$$

Here $q_{b_i} > a_{q_j}$

So $(p_{b_i} - a_{p_j}) a_{q_j} < (p_{b_i} - a_{p_{\text{min}}}) q_{b_i}$.

All other assignments with higher price or with smaller quantity generates smaller net surplus. This assignment results in maximum improvement in objective function if the sizes of ask and bid are same. So $R_{b_{\text{imax}}}$ is the maximum improvement that can be done in any assignment in objective function.

Improvements with two attributes price and size: Let $c_{ov_{ik}}$ be the contribution to value of objective function for assigning quantity $b_{q_i}$ of $A_k$ to $B_i$ is given by,

$$c_{ov_{ik}} = (b_{p_i} - a_{p_k}) b_{q_i} - w_{ik} b_{q_i}$$

In this case both terms are always nonnegative. Since $b_{p_i} \geq a_{p_k}$ and wastage penalty is non-negative. The second term is subtracted from first term.
So, the maximum contribution occurs when the second term is minimum i.e. 0. So, the assignment that gives the maximum contribution must have minimum ask price and same value for size attributes. Let $A_l$ be the ask with minimum ask price $a_{p_l}$ (also called as $a_{p_{min}}$) and its size be same as $b_{s_l}$. The contribution to the value of objective function is

$$\text{cov}_{i l} = (b_{p_i} - a_{p_{min}})b_{q_l} - (b_{p_i} - a_{p_{min}})b_{q_l}$$

(3.20).

This will be the maximum contribution, since for any other ask $A_k$,

$$(b_{p_i} - a_{p_{min}})b_{q_l} \geq (b_{p_i} - a_{p_k})b_{q_l} \text{ and } w_{ik}b_{q_l} \geq 0$$

It can be seen that if price remains same and ask size increases then contribution decreases. If price remains same and ask size decreases, then contribution improves.

Let $\text{cov}_{ik}$ be the contribution to value of objective function for assigning quantity $b_{q_i}$ of $A_k$ to $B_i$. In the same way let $\text{cov}_{ij}$ be the contribution for assigning quantity $b_{q_i}$ of ask $A_j$ to $B_i$. Then they are given by

$$\text{cov}_{ik} = (b_{p_i} - a_{p_k})b_{q_i} - w_{ik}b_{q_i}$$

(3.21).

$$\text{cov}_{ij} = (b_{p_i} - a_{p_j})b_{q_i} - w_{ij}b_{q_i}$$

(3.22).

Then assignment of $j^{th}$ ask is an improvement, if and only if,

$$\text{cov}_{ij} - \text{cov}_{ik} > 0$$

$$(b_{p_i} - a_{p_j})b_{q_i} - w_{ij}b_{q_i} - ((b_{p_i} - a_{p_k})b_{q_i} - w_{ik}b_{q_i}) > 0$$

$$(b_{p_i} - a_{p_j}) - w_{ij} - (b_{p_i} - a_{p_k}) + w_{ik} > 0$$

$$b_{p_i} - a_{p_j} - w_{ij} - b_{p_i} + a_{p_k} + w_{ik} > 0$$

$$a_{p_k} - a_{p_j} + w_{ik} - w_{ij} > 0$$

$$(a_{p_k} - a_{p_j}) > (w_{ij} - w_{ik})$$

(3.23).

So the assignment of $j^{th}$ ask improves the value of objective function, if condition (3.23) is satisfied. This condition is used in algorithm to find out whether any assignment improves the objective function.

### 3.4 Optimum Assignment for Set of Bids

In this analysis it is assumed that, all the assignments are feasible. Let $AS$ be the set of all asks. Let $B_1$ and $B_2$ be two bids. Let $AS_1$ be the set of asks which are assigned to bid $B_1$. Let the contribution to the value of objective function $\text{cov}_1$, from this assignment be
the maximum. So there is no ask $A_i$ or set of asks $AS_i, \epsilon (AS-AS_1)$, such that its contribution $cov_1 > cov_1$. The contribution cannot be improved by changing the quantity assigned. Let $bq_1$ be the quantity assigned at this stage.

In the same way $AS_2$ be the set of asks assigned to bid $B_2$. This set $AS_2$ may contain some asks from $AS_1$, whose supply is not completely satisfied. Let the contribution to the value of objective function $cov_2$, from this assignment be the maximum. This assignment satisfies the property that, there is no ask $A_i$ or set of asks $AS_i, \epsilon (AS-AS_1-AS_2)$, such that its contribution (when assigned to bid $B_2$) $cov_1 > cov_2$. The contribution cannot be improved by changing the quantity assigned. Let $bq_1 + bq_2$ be the quantity assigned at this stage.

In this case $cov_1$ is the optimum contribution, which cannot be improved. On the other assignment to bid $B_2$ is optimum among the currently unassigned asks (i.e. not assigned to bid $B_1$ or assigned to $B_1$ with unfulfilled supply). The contribution $cov_2$ may be improved by asks from $AS_1$. In this case improvement means assignment of asks with higher unit contribution or assignment of higher quantity from asks assigned to bid $B_1$. So when we refer to improvement or higher contribution, both the cases are considered.

Then the assignment of all asks in $AS_1$ to $B_1$ and $AS_2$ to $B_2$ with value of objective function $cov_1 + cov_2$ is optimum in following two cases.

(a) If some or all the asks which are assigned to $B_1$, can be assigned to $B_2$ and vice versa and this change does not involve any ask which is not in $(AS_1 \cup AS_2)$, then assignment $A_1$ to $B_1$ and $A_2$ to $B_2$ is the optimum for both bids together. The quantity replaced is the same. In this case there is no change in total quantity assigned ($bq_1+bq_2$). This follows from interchange theorem.

(b) If no ask can be assigned to both bids $B_1$ and $B_2$, (there is no common set of asks which can be assigned to both the bids) then above assignment is optimum. As there is no assignment, which improves the value of objective function. In this case $(AS_1 \cap AS_2) = \phi$.

In case (b), both $cov_1$ and $cov_2$ cannot be improved so $cov_1 + cov_2$ cannot be improved. In the same way in case (a), $cov_1 + cov_2$ cannot be improved.
We now consider the case where, an ask or set of asks, which improve contribution \( \text{cov}_2 \) to the value of objective function are assigned to bid \( B_1 \). (This includes the case where \( \text{AS}_2 \) can be an empty set.) We do not consider the case of bid \( B_1 \) as \( \text{cov}_1 \) is globally optimum. So it cannot be improved. Let \( \text{AS}_{2g} \) be the set of asks currently assigned to bid \( B_1 \) and have higher contribution than asks in \( \text{AS}_2 \), to the value of objective function from assignment to bid \( B_2 \). This set \( \text{AS}_{2g} \) contains, set of asks assigned to \( B_1 \) with higher unit contribution, as well as asks currently partly assigned to both bids \( B_1 \) and \( B_2 \), where higher quantity can be assigned to bid \( B_2 \) by decreasing quantity assigned to bid \( B_1 \), if such an assignment improves contribution from bid \( B_2 \).

Let \( \text{AS}_{2n} \) be the set of asks, with the highest contribution \( \text{cov}_{2n} \) to the value of objective function from assignment to bid \( B_2 \). In other words \( \text{cov}_{2n} > \text{cov}_2 \) and there is no ask \( A_i \) or set of asks \( \text{AS}_i \) in \( (\text{AS} - \text{AS}_{2n}) \), which improves \( \text{cov}_{2n} \). Let \( \text{AS}_4 \) be the set of asks from \( \text{AS}_2 \), replaced in new assignment. So \( \text{AS}_{2n} \) can be written as \( \text{AS}_{2n} = (\text{AS}_2 - \text{AS}_4) \cup \text{AS}_{2g} \).

Let \( \text{AS}_{1n} \) be the set of asks from currently unassigned and unfulfilled asks, with contribution to the value of objective function \( \text{cov}_{1n} \) from assignment to bid \( B_1 \). Suppose that apart from sets \( \text{AS}_1 \), this is the next highest contribution. In other words \( \text{cov}_1 > \text{cov}_{1n} \), however, there is no ask or set of asks in \( (\text{AS} - \text{AS}_{2n}) \), which improves it. Further \( \text{AS}_{1n} \) contains some asks from set \( (\text{AS} - (\text{AS}_1 \cup \text{AS}_2)) \).

Then the optimum assignment can be determined as follows.

(c) If \( \text{cov}_1 + \text{cov}_2 \geq \text{cov}_{1n} + \text{cov}_{2n} \), then the first assignment (\( \text{AS}_1 \) to bid \( B_1 \) and \( \text{AS}_2 \) to bid \( B_2 \)) is optimum.

(d) If \( \text{cov}_1 + \text{cov}_2 < (\text{cov}_{1n} + \text{cov}_{2n}) \) then the second assignment (\( \text{AS}_{1n} \) to bid \( B_1 \) and \( \text{AS}_{2n} \) to bid \( B_2 \)) is optimum. The second assignment is obtained by repeated replacement.

This can be easily generalized for set of \( k \) bids. Suppose that \( \text{AS}_{k-1} \) be the set of asks assigned to set \( \text{BD}_{k-1} = \{B_1, B_2, \ldots, B_{k-1}\} \) of \( (k-1) \) bids. Let \( \text{cov}_{k-1} \) be the optimum contribution. Let \( \text{AS}_k \) be the set of asks assigned to bid \( B_k \). This assignment is currently
optimum for bid $B_k$. Let $cov_k$ be the contribution to the value of objective function from this assignment. Then optimum assignment for set $k$ bids can be obtained as follows.

1. Determine set of asks which can be assigned to bid $B_k$ (currently assigned to set $k-1$ bids) and satisfy (i) have higher unit contribution (ii) assignment of quantity to bid $B_k$ can be increased.

2. Then find out which condition (a) or (b) is satisfied. In these cases, we have the optimum assignment.

3. In other cases, we work optimum assignment as in case (c) and (d). Determine the set of asks $AS_{kg}$, which is currently assigned to set of bids $BD_{k-1}$, which can improve contribution (higher unit contribution as well as quantity) to the value of objective function by assignment to bid $B_k$. Then determine set of asks $AS_{kn}$ with the highest possible contribution $cov_{kn}$ to the value of objective function from assignment to bid $B_k$. So there is no ask or asks in $(AS - AS_{kn})$, which can improve $cov_{kn}$. Then determine set of asks with $AS_{k-1n}$, which can be assigned to first $(k-1)$ bids and have the next highest contribution.

4. Then determine the optimum assignment for set $BD_k = \{B_1, B_2, ..., B_k\}$ by applying conditions (c) and (d). So if $(cov_{k-1} + cov_k \geq cov_{k-1n} + cov_{kn})$, then assignment $AS_{k-1}$ to bid $BD_{k-1}$ and $AS_k$ to bid $B_k$ is optimum. Otherwise assignment $AS_{k-1n}$ to bid $BD_{k-1}$ and $AS_{kn}$ to $B_k$ is optimum for $BD_k$.

Stage-wise improvement: The value of objective function can be improved either by increasing surplus from an assignment without increasing wastage factor or by an assignment which brings down wastage without increasing surplus by corresponding amount. The optimum assignment is obtained in stages. In first stage an optimum assignment for a single bid. Then an optimum assignment for set of two bids is determined by carrying out steps described earlier in this section. This procedure is repeated till all assignments are completed. Thus at each stage we obtain the optimum assignment. We present two algorithms for generating optimum assignment in case of assignment and indivisibility constraints. First, we present algorithm to obtain optimum solution where there are only assignment constraints. In the second case, we present an algorithm, which can handle bids with indivisible demand constraints. In the second
algorithm, there is an assignment tree construction, which helps in determining how the quantity can be reassigned without changing the value of objective function.

### 3.5 Algorithm for Obtaining Optimum Assignment

Let the terms \( b_p \), \( a_p \) and \( w_{ij} \), be as defined earlier. If \( i^{th} \) bid can be assigned to \( j^{th} \) ask and \( b_p_i \geq a_p_j \), then it is called as feasible assignment. In our algorithm, in first stage, a table of feasible assignments is created. This table consists of three elements. If \( i^{th} \) bid can be assigned to \( j^{th} \) ask and \( b_p_i \geq a_p_j \), then three elements of the table are \( i \) (called as bid indicator), \( j \) (called as ask indicator) and unit contribution resulting from assignment of \( i^{th} \) bid and \( j^{th} \) ask. The last element is

\[
ucov_{ij} = (b_p_i - a_p_j) - w_{ij}
\]

In case \( i_1^{th} \) bid cannot be assigned \( i_2^{th} \) ask or \( b_p_{i_1} \geq a_p_{i_2} \), then entry \( i_1, i_2, ucov_{i_1i_2} \), is absent from the table. So all feasible assignments can be determined by scanning this table.

We start assignment from the bid with the highest contribution/the highest unit contribution. Then, an ask or a set of asks, which can be assigned to current bid and have the highest possible unit contribution for this bid is determined. In the next step the maximum quantity that can be assigned is obtained. This assignment maximizes the value of objective function i.e. net surplus and minimum wastage by theorem 6. If ask with maximum unit contribution has enough supply to satisfy the demand of highest price bid, then bid and ask are marked as assigned and assignment continues from next bid. If ask quantity is less than bid quantity, then the ask is marked as temporarily assigned, while bid is marked as partly assigned. In case bid quantity is more, the assignment is continued till demand is satisfied. In the next step, assignment continues from the bid with next highest contribution/the highest unit contribution. First an ask or set of asks from unassigned asks with the highest contribution is determined. Then the optimum assignment is determined as discussed in section 3.4. The process is continued till complete either demand is fulfilled or supply is exhausted. At every stage we have an optimum assignment for set of bids and an assignment, which is optimum for current bid. The optimum assignment for set of bids and current bid is determined. This is the maximum possible improvement at any stage. This follows from the results of theorems.
1, 2, 3, 4 and 6 depending upon the assignment. The bid and all asks are marked as assigned. Then assignment is continued in decreasing order of unit contribution. If ask has more quantity, then bid is marked as assigned and ask is marked as partly assigned.

Initially a table indicating demand for different width is constructed. If the ask size is multiple of bid size (e.g. bid width 800 cm and ask width 1600 cm) then this table is used to see whether there is demand for remaining width (i.e. 800 cm). If demand is there then wastage is set to 0. This is helpful in situations to decide, whether bid is to be matched with ask 1600 cm or 1000 cm. Here matching of current bid with an ask with width of 1000 cm will show lesser wastage than that of 1600 cm. However, if there is a demand for 800 cm then it can be matched with the ask of 1600 cm width so that wastage is minimized.

The assignment is continued till one of the three conditions holds good:

(i) no ask is left
(ii) no bid is left
(iii) either total supply is exhausted or total demand is fulfilled.

In our solution assignment is carried out if bid price is either more than ask price or both are same. This assumption is reasonable in the sense that in most exchanges asks are cleared with bids of same or higher price. It can also be seen in [KDL2001], that equilibrium price is first obtained. This price is used to determine the asks and bids which can be cleared (asks below this price and bids above it). Let AS be the list of asks and BD be the list of bids. The main algorithm can be seen in figure 3.1.

```
Algorithm findopasg(AS, BD)
    Call Create_Unit_Contr_Table
    Call Create_Size_Demand_Table(bids);
    While (there is unassigned bid in B) {
        Call get_next_unassigned_bid(bids);
        Call get_opt_ask(bid_quantity, bid_size,);
        Call assignment(bid, ask, bid_quantity);
    }
    return;
```

Figure. 3.1 Main algorithm

This algorithm calls function “Create_Size_Demand_Table”. This function creates size demand table, which indicates the demand for different values of size attribute. The usage of this table has already been explained. This function can be seen in figure 3.2.
It then calls function “get_next_unassigned_bid()”. This function returns the next unassigned bid. This function can be seen in figure 3.3.

The function “get_opt_ask()” returns the set of asks which bring the maximum improvement in the value of objective function. This function in turn calls “get_next_unassigned_ask()”, which returns the next unassigned ask that can be assigned to current bid. Then it determines optimum assignment by finding out whether there are assigned asks with higher contribution, which can be assigned to current bid. It then calls function “cal_obv()” to find out improvement in the value of objective function.

Then it selects assignment with maximum value. This function also calls function “search_table()”, which searches the size demand table, for a particular value of width attribute, which is passed on as a parameter and returns true, if demand for that value exists. If it returns “true”, then wastage is set to 0. These functions can be seen in figure 3.4. The function “assignment()” carries out the assignment of asks and bids and marks them appropriately. This function can be seen in figure 3.5.

```
Create_Size_Demand_Table(bids) {
    while ( there is next bid) {
        get bid_size, bid_quantity;
        call search_table(bid_size,table_size);
        if (.not. found ) then {
            increase current_index by 1;
            store bid_size to search_table(current_index,1);
            store bid_quantity to search_table(current_index,2); set table_size to current_index ;}
        else {
            add bid_quantity to search_table(current_index,2)}
    read next bid ; }
return ;
}
```

```
Create_Unit_Contr_Table(bids) {
    while ( there is next bid) {
        while (there is next ask) {
            get bid_size, bid_price;
            get ask_price,ask_size ;
            if ( bid_price ≥ ask_price ) then {
                increase current_index by 1 ;
                unit_contr = ( bid_price – ask_price ) – wastage_penalty(ask_size,bid_size) ;
                store bid_index,ask_index, unit_contr to contr_table(current_index,1) ;
            };
        }
    read next bid ; }
return ;
}
```

```
get_next_unassigned_bid(bids) {
    while ( there is unassigned bid) {
        if (bid_price > max_price) then {
            set max_price to bid_price;
            set bid to current_bid ;
        }
    return ;
}
```

In this algorithm for any bid, we always determine the set of asks, which bring the maximum improvement in the value of objective function. Then it determines the maximum quantity that can be assigned. Then assignment is carried out. This process is repeated for each ask and bid.
Example 3.1: The working of the algorithm is illustrated in the following example. There are five asks and five bids. The assignment constraints are basically size constraints i.e. a bid can be matched with an ask of same or higher size. The wastage penalty is 8 here. The output can be seen in Table – 4.4.

The algorithm has been implemented in C++. It is tested with randomly generated data sets of different sizes. Each data set consisted of ask price, ask quantity, ask size, bid price and bid quantity. The data set consisted of ask price, quantity, ask size, bid size, bid price and bid quantity. Size of data sets varied from 5 to 100.

---

**Algorithm**

- **get_opt_ask(bid_quantity, bid_size)**
  
  ```
  while (there is unassigned ask and asg_quantity < bid_quantity)
  call get_next_unassigned_ask(bid_index);
  if (ask_quantity >= bid_quantity) asg_quantity = ask_quantity + bid_quantity;
  if (bid_quantity > ask_quantity) then qty_asg = asg_quantity + ask_quantity;
  if (optimum return);
  copy assignment to old_assignment;
  ```

- **get_next_unassigned_ask(bid_index)**
  
  ```
  while (i <= table_size)
  if (ask unassigned and bid_index = bid) and (unit_contr > ucontr)
  { ucontr = unit_contr; current_ask = ask_index; qty_sup = qty_sup + ask_qty; }
  if (ask assigned and bid index = bid) { qty_inv = qty_inv + qty_assigned; }
  if (qty_inv = 0) or (qty_inv >= ask_qty) {optimum = true; return }
  ```

- **cal_obv(bid, ask, qty_asg)**
  
  ```
  net_surplus = (bid_price – ask_price) * bid_quantity;
  wastage = (ask_size – bid_size) * bid_quantity;
  call search_table(wastage, table_size);
  if found then set wastage = 0; ov = net_surplus + wastage; return 
  ```

- **Search_Table(bid_size, table_size)**
  
  ```
  while (i <= table_size)
  { if (search_table(i, 1) = bid_size) then {set current_index to i; return .true; }
  else {increase i by 1; } return false; }
  ```

---

Figure. 3.4 Functions used in main algorithm findopasg
The results were compared with unconditional optimum solution and some solutions obtained with the help of the MATLAB package. It has been observed that our algorithm always generated optimum solution. It has also been seen that if the size of ask is constant and the bid sizes are variable but take few values (as in most of the practical cases), then we can ignore the wastage factor. The approach can be, to obtain the maximum surplus and then readjust assignment without affecting value of objective function. It can also be seen that time complexity of our algorithm is always polynomial.

In this algorithm a matching ask which generates the maximum improvement can be obtained by scanning unassigned asks at any point of time. In the first instance there are \( n \) unassigned asks, in the next instance there are \( (n-1) \) unassigned asks and so on. So, in all, the solution will require to scan \( n(n-1)/2 \) asks. So, the time complexity is polynomial and of the order \( O(n^2) \) or \( O(mn) \). The unit contribution table created with time complexity is \( O(nm) \). So the time complexity is always polynomial. Apart from this, the demand size table and minimum or maximum price asks/bids can be obtained with linear time complexity. So, overall, the time complexity is always polynomial.

<table>
<thead>
<tr>
<th>Bid</th>
<th>Price</th>
<th>Quantity</th>
<th>Size</th>
<th>Ask Price</th>
<th>Quantity</th>
<th>Size</th>
<th>Bid Price</th>
<th>Ask Spread</th>
<th>Quantity</th>
<th>Net Surplus</th>
<th>Wastage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>187</td>
<td>11</td>
<td>8</td>
<td>101</td>
<td>23</td>
<td>8</td>
<td>1</td>
<td>86</td>
<td>11</td>
<td>946</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>181</td>
<td>12</td>
<td>4</td>
<td>109</td>
<td>8</td>
<td>12</td>
<td>2</td>
<td>80</td>
<td>12</td>
<td>960</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>173</td>
<td>23</td>
<td>8</td>
<td>3</td>
<td>121</td>
<td>6</td>
<td>12</td>
<td>38</td>
<td>4</td>
<td>152</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>161</td>
<td>10</td>
<td>12</td>
<td>135</td>
<td>4</td>
<td>8</td>
<td>3</td>
<td>22</td>
<td>19</td>
<td>418</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>157</td>
<td>8</td>
<td>8</td>
<td>151</td>
<td>23</td>
<td>8</td>
<td>4</td>
<td>52</td>
<td>8</td>
<td>416</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>40</td>
<td>2</td>
<td>80</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>36</td>
<td>4</td>
<td>144</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>64</td>
<td>3140</td>
<td></td>
<td></td>
<td></td>
<td>64</td>
<td>8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure. 3.5 Function Assignment
This compares quite favorably with the complexity $O((nm + n^2 \log n))$ of algorithm presented in [KDL2001], apart from its simplicity for large instance of $n$ and $m$. The comparison can be seen in the figure – 3.6.

Algorithm for assignment in case of indivisible demand constraints: Now the algorithm to obtain the optimum assignment in case of indivisibility constraints is presented. This algorithm can also handle other types of assignment constraints. This algorithm differs from above algorithm, since it needs to keep track of quantity assigned in case of bids with indivisible demand constraints. It uses assignment tree construction to determine a set of bids to which assignment should be carried out without changing the value of objective function.

**Assignment Tree**

The algorithm uses assignment tree data structure to determine the assignment of asks and bids, which do not change the value of objective function. Assignment tree helps in determining distribution of ask quantity among different bids, so that the value of objective functions remains unchanged from unconstrained assignment and bids with indivisible demand are satisfied.

![Figure. 3.6 Comparative Performance of the Algorithm (assignment constraints)](image)
This tree is constructed when an ask has more supply than the current bid or to obtain adjustment quantity for combining different bids. The left child of assignment tree always represents bids without indivisible demand bid constraints and right child represents the bids with indivisible demand bid constraints, which can be satisfied by current ask. The nodes at first level are the ones which bring highest improvement, nodes at second level are the next highest and so on. All the paths represent different assignments of the current ask. Any one of the assignments can be selected, as they do not change the value of objective function.

The algorithm selects one assignment (application of theorem 1 to 5 as the case may be). This is helpful in cases where only one ask can be assigned to bids with indivisible demand bid constraint. It can be seen, by applying results from theorem 1 and 2, that the different assignments obtained from assignment tree, will not change the value of objective function. So, these paths can be used to determine the assignment. In the same way while assigning asks to bids, which require combining of different bids, the assignment tree is constructed to obtain adjustment quantity, especially when the next bid has indivisible demand bid constraint. Using this adjustment quantity the asks are distributed among these bids. In this case, it can be seen, by applying results from theorems 1, 2, 3, 4 and 5 that, redistribution of quantity among different bids, will not change the value of objective function. An assignment tree is shown in figure 3.7, which indicates two different situations of assignment tree.

These constructs ensure that it is possible to redistribute quantity among different bids without affecting objective function value. The adjustment scenario is shown in Figure 3.7(a). The current bid has demand of 3 and the next bid has indivisible demand constraint with demand of 10. The current ask has supply of 12. The assignment of 10 from current ask to next bid, 2 to current bid and assignment of 1 from next ask do not change the value of objective function. Its value remains the same as unconstrained assignment. Figure 3.7(b) shows where supply is more than demand. In this case
assignment of current ask to bids 1 and 3, or 1 and 4, do not change the value of objective function. This follows from application of theorem 1,2,3,4 and 5.

We give brief description of working of this algorithm. The assignment starts from the bid with the highest contribution/the highest unit contribution. This bid is assigned to ask which maximizes the value of objective function i.e. net surplus and minimum wastage penalty. If this bid has indivisible demand bid constraint then an ask of same or more quantity which brings the maximum improvement in the value of objective function is selected. If the ask quantity is less than the bid quantity and bid does not have indivisible demand bid constraint, then the ask is marked as assigned, while bid is marked as partly assigned and will continue the assignment from next ask onwards.

If both quantities are same then asks and bids are marked as assigned and the assignment continues. The assignment is continued in decreasing order of unit contribution for bids. At each stage the bid is assigned to currently available ask or set of asks which gives the maximum improvement in value of objective function (theorem 6). In case of bids with indivisible demand bid constraint, we obtain an ask of equal or more quantity that brings
the maximum improvement (theorem 6). When ask has more quantity, then an assignment tree is constructed. Using this assignment tree, an assignment is worked out. The assignment tree is also constructed while combining different bids for finding out adjustment quantity. In this case we apply theorems 4 and 5. After determining this assignment, optimum assignment is worked as shown in section 3.4. A table indicating demand for different width is constructed. Its usage is already explained. The assignment is continued till one of the following three conditions hold good (i) no ask is left (ii) no bid is left (iii) ask price exceeds that of bid price.

In this solution, assignment is carried out if bid price is either more than ask price or if both are same. This assumption is reasonable in the sense that in most exchanges asks are cleared with bids of same or higher price. It can also be seen in [KDL2001], that equilibrium price is first obtained. This price is used to determine the asks and bids which can be cleared (asks below this price and bids above it). Let AS be the list of asks and BD be the list of bids.

Algorithm findopasgic(AS,BD) presented here generates the optimum assignment of asks and bids which maximizes surplus and minimizes the wastage, while satisfying indivisible demand and assignment constraints. As in earlier algorithm the table of unit contributions where assignments are feasible is created. Then it creates the demand size table, which contains demand for different sizes. It uses number of functions for finding ask which generates the maximum improvement in the value of objective function, assigning asks to bids and for assignment tree construction. The main algorithm can be seen in figure 3.8.

```
Algorithm findopasgic(AS,BD)
    Call Create_Unit_Contr_Table
    call create_size_demand_table(bids);
    while (there is unassigned bid in B) {
        if (nobid = false) then get next unassigned bid;
        call get_opt_ask(bid_quantity,bid_size,bid_type); if(bid_size=ask_size) then
            call assignment(bid,ask,bid_quantity);
            if (ask_size > bid_size) then
                call find_no_change_assignment(bid,ask); if (bid_size > ask_size and bid_divisible = True) then
                call more_assignment(bid,ask); check_optim(assignment) return;
        }
    }
```

Figure. 3.8 Main algorithms

It calls function “get_opt_ask()”, which finds out the ask which brings out maximum improvement in the value of objective function from current assignment. This function can be seen in figure 3.9.
get_opt_ask(bid_quantity,bid_size,bid_type)

\[
\begin{align*}
\text{do while ( there is unassigned ask) } & \{ \text{ read next unassigned_ask ;} \\
\text{ if ( bid_type = indivisible) } & \text{ then qty_asg = bid_quantity ;} \\
\text{ if ( ask = unassigned and ask_quantity } & \geq \text{ bid_quantity and ask_size } \geq \text{ bid_size) then} \\
\text{ qty_asg = bid_quantity ; call cal_obv(bid,ask,qty_asg) ;} \\
\text{ if ( bid_type = divisible) and ( ask_size } & \geq \text{ bid_size ) ) then } \\
\{ \text{ if ( bid_quantity } < \text{ ask_quantity) } & \text{ then } \{ \text{ qty_asg = bid_quantity ;} \} \}\text{ else } \{ \text{ qty_asg = ask_quantity ; } \} \text{ call cal_obv(bid,ask,qty_asg) ;} \\
\text{ if ( ov } > \text{ max_imp) then } & \{ \text{ ask_ret = current_ask ; max_imp = ov ; } \} \text{ return ; } \}
\end{align*}
\]

Figure. 3.9  Function to get optimum ask

It also calls function “create_size_demand_table()”, which creates the size demand table. This function stores the total demand for different values of width attribute. The function “get_opt_ask()”, calls another function “cal_obv()”, which finds value of objective function after current assignment and returns ask/set of asks which bring maximum improvement. While finding value of objective function it uses “search_table()”, which searches the size demand table for a particular value of width (which is passed as parameter) and returns true, if demand for that value exists. If “true” is returned then wastage is set to 0. These functions are shown in figure 3.10. The function “no_change_assignment()”, constructs assignment tree as described earlier for the current ask. It calls function “get_next_bid()”, to find out bids, which can become nodes of assignment tree.

These are the bids, to which assignment can be made without changing value of objective function. The function “more_assignment()”, is used to set “nobid” parameter to true, when bid quantity is more than ask quantity. When value of this parameter is true, algorithm continues assignment of current bid. Main algorithm and other functions use function “assignment()”, to assign asks to bids. If the quantity of assignment is equal, then asks and bids are marked as fully assigned otherwise they are marked as partly assigned depending upon the quantity. It also adjusts width parameter.

The function check_optim() finds out whether the assignment is optimum or not. It is done by finding out whether any assignment of assigned ask or set of asks with higher
unit contributions improves current value of objective function. In case there is improvement new assignment is selected as optimum assignment. These functions are shown in figure 3.11. This algorithm first locates ask/set of asks that brings out maximum improvement in value of objective function, effects assignment and repeats the process. The working of algorithm is illustrated with following simple example.

Example 3.2: In Table 3.5 a simple example with 5 bids and 4 asks is shown. Out of these five bids, first three bids have indivisible demand bid constraints. The table is divided into three parts which show bids submitted by buyers, asks submitted by sellers
and the optimum assignment generated by our algorithm. It generates the net surplus of 1024 without any wastage, which is same as optimum solution.
3.6 Experimental Results and Conclusion

The algorithm is implemented in C++. It is tested with randomly generated data sets of different sizes. Each data set consisted of ask price, ask quantity, ask size, bid size, bid price and bid quantity. The sizes of data sets varied from 5 to 100. The results were compared with unconditional optimum solution and some solutions obtained with the help of the MATLAB package. The performance of our algorithm against other algorithms [KDL2001] for number of different bids is plotted in figure 3.12. It shows the number of asks and bids required to be scanned by our algorithm against the other

Table. 3.5 - Illustrative example of solution generated by algorithm

<table>
<thead>
<tr>
<th>Bids</th>
<th>Asks</th>
<th>Assignment obtained by Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bid</td>
<td>Price</td>
<td>Quantity</td>
</tr>
<tr>
<td>1</td>
<td>171</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>167</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>161</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>154</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>151</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>33</td>
<td></td>
</tr>
</tbody>
</table>

Figure. 3.12 Comparative performance of algorithm

algorithms. In the graph, x-axis represents number of bids and y-axis represents number of asks and bids required to be scanned to obtain the optimum solution. The time
complexity of our algorithm is polynomial. An ask/set of asks which can give maximum improvement can be obtained by scanning at most \( n \) asks.

**Theorem 7:** It can be seen that the algorithm will generate optimum solution in both cases. (Indivisibility constraints with or without assignment constraints and only assignment constraints).

**Proof:** Let \( AS \) be the set of all asks. Let \( AS_1 \) be the set of ask/asks, which when assigned to bid \( B_1 \), brings the maximum improvement in the value of objective function. Let \( \text{cov}_{a1} \) be the contribution value of objective function after assignment of these asks/ask to bid \( B_1 \). If bid has indivisible bid constraint then there will be only one element in \( AS_1 \). There is no ask \( A_i \) or set of asks \( AS_i \in (AS - AS_1) \), such that \( \text{cov}_{a1} \) can be improved. The contribution cannot be improved by changing quantity assigned in respect of any ask in \( AS_1 \). At this stage \( bq_1 \) be the quantity assigned.

Let \( AS_2 \) be the set of ask/asks, which brings maximum improvement in the value of objective function after assignment to second bid \( B_2 \). This set will not contain set of asks in \( AS_1 \), whose supply is completely satisfied but may contain some asks in \( AS_1 \), which are partly assigned to first bid. Let \( \text{cov}_{a2} \) be the contribution to the value of objective function from this assignment. The value of \( \text{cov}_{a2} \) is 0, when \( AS_2 \) is empty set. There is no ask \( A_i \) or set of asks \( AS_i \in (AS - AS_1 - AS_2) \), such that \( \text{cov}_{a2} \) can be improved. The quantity assigned is maximum possible. Let \( OV_2 \) be the value of objective function after second assignment, then clearly that is the maximum possible value. The contribution cannot be improved by changing quantity assigned in respect of any ask in \( AS_2 \). At this stage \( (bq_1 + bq_2) \) be the quantity assigned.

It can be seen that \( OV_2 = \text{cov}_{a1} + \text{cov}_{a2} \). 

\[(3.24)\]

Let \( \text{cov}_{a1i} \) be the highest contribution to the value of objective function from assignment of asks \( \epsilon (AS - AS_1) \) to bid \( B_1 \). Then we have \( \text{cov}_{a1} > \text{cov}_{a1i} \).

Let \( \text{cov}_{a2i} \) be the highest contribution to the value of objective function from assignment of asks \( \epsilon (AS - AS_1 - AS_2) \) to bid \( B_2 \). Then we have \( \text{cov}_{a2} > \text{cov}_{a2i} \).

So we have \( OV_2 = \text{cov}_{a1} + \text{cov}_{a2} > \text{cov}_{a1i} + \text{cov}_{a2i} \). 

\[(3.25)\]
It is possible that assignment of asks from set AS\(_1\) may improve the contribution from second bid B\(_2\). Three possible cases here are as follows.

1. No common set of asks between bids B\(_1\) and B\(_2\).
2. It is possible that assignment of asks from set AS\(_1\) may improve the contribution from second bid B\(_2\). However these asks can be completely replaced by asks in AS\(_2\) and can be assigned to bid B\(_1\). In this case only asks from (AS\(_1\) \cup AS\(_2\)) are assigned to bids B\(_1\) and B\(_2\). The total quantity assigned is (bq\(_1\)+bq\(_2\)).
3. An ask or set of asks in AS\(_1\) (currently assigned to bid B\(_1\)) can improve contribution to the value of objective function, if they are assigned to bid B\(_2\) (both cases i.e. asks assigned to bid B\(_1\) with higher unit contribution or assignment of higher quantity than currently assigned to bid B\(_2\) in case of asks assigned to both bids are considered).

However this requires assignment of asks, currently not in (AS\(_1\) \cup AS\(_2\)) to bid B\(_1\) (in order to satisfy demand of B\(_1\) and B\(_2\)). In other words it requires assignments of asks not in (AS\(_1\) \cup AS\(_2\)). In some cases the total quantity assigned can be smaller than (bq\(_1\)+bq\(_2\)).

In case (1), it can be seen that both terms of expression (3.24) i.e. cov\(_{a1}\) and cov\(_{a2}\) cannot be improved from asks \(\epsilon (AS - AS\(_1\) - AS\(_2\))\). Further no ask from AS\(_1\) can be assigned to bid B\(_2\) and no ask from AS\(_2\) can be assigned to bid B\(_1\). So the value of objective function cannot be improved. Thus expression (3.25) is the optimum value.

In case (2), the current value of objective function is OV\(_2\) = cov\(_{a1}\) + cov\(_{a2}\). Here cov\(_{a1}\) cannot be improved. The term cov\(_{a2}\) can be improved by asks currently assigned to bid B\(_1\). However it is possible to assign asks from AS\(_2\) to B\(_1\), to replace the asks now assigned to bid B\(_2\). This will not change the value of objective function by interchange theorem. So (3.25) is the optimum value, which cannot be improved.

In case (3) assignment of asks in AS\(_1\) to bid B\(_2\) decreases the contribution to the value of objective function from B\(_1\). Let AS\(_3\) be the set of asks (currently not in AS\(_1\) or AS\(_2\)) which replaces the asks in AS\(_1\) with higher unit contribution (or assignment of higher quantity) to bid B\(_2\). (The asks from AS\(_3\) are now assigned to bid B\(_1\)). Suppose that this assignment results in the next highest contribution cov\(_{a1}\) to bid B\(_1\). In other words cov\(_{a1}\) \(\geq\) cov\(_{a1}\). The quantity assigned to bid B\(_1\) is the maximum possible at this stage. There is no unassigned ask, which can produce higher contribution than cov\(_{a1}\).
Let $\text{cov}_{n2}$ be the highest contribution to the value of objective function from new assignment to bid $B_2$. So the value of objective function is

$$\text{OV}_{n2} = \text{cov}_{n1} + \text{cov}_{n2}.$$ 

In this case second value cannot be improved. If $\text{cov}_{a1} + \text{cov}_{a2} > \text{cov}_{n1} + \text{cov}_{n2}$, then first assignment is optimum. This follows because no other assignment of asks to bids $B_1$ and $B_2$ improves the value of objective function. There is no ask or set of asks which can improve $\text{cov}_{a1}$. The quantity assigned is the maximum possible. Only ask or set of asks assigned to bid $B_1$ improves the contribution $\text{cov}_{a2}$. However this change does not improve the value of objective function as earlier value of objective function is higher.

In case $\text{cov}_{a1} + \text{cov}_{a2} < \text{cov}_{n1} + \text{cov}_{n2}$, the second assignment is optimum. This follows because contribution ($\text{cov}_{n1} + \text{cov}_{n2}$), cannot be improved.

Here $\text{cov}_{a1} > \text{cov}_{n1}$ and $\text{cov}_{a2} < \text{cov}_{n2}$, such that $\text{cov}_{a1} + \text{cov}_{a2} < \text{cov}_{n1} + \text{cov}_{n2}$.

The contribution $\text{cov}_{a2}$ cannot be improved. The first term $\text{cov}_{a1}$ can be improved, but this results in decrease in value of second term so that total value is not optimum i.e. $\text{cov}_{a1} + \text{cov}_{a2} < \text{cov}_{n1} + \text{cov}_{n2}$. Any other assignment decreases the value of objective function. So the second assignment is optimum.

So for $k = 1$ and $k = 2$ the result is optimum. Suppose that after assignment to $k$ bids $B_1$, $B_2$, $\ldots$, $B_k$ the value of objective function is $\text{OV}_k$. Let this be the optimum value. Let $\text{AS}_k$ be the set of asks which are assigned to all these $k$ bids. Let $\text{AS}_{k+1}$ be the set of unassigned asks (remaining ask or set of asks from $\text{AS}_k$, which are not completely assigned) which has the highest contribution to the value of objective function after assignment to bid $B_{k+1}$. Let $\text{cov}_{ak+1}$ be this contribution. Then the value of objective function is $\text{O}_{k+1} = \text{O}_k + \text{cov}_{ak+1}$.

This value can be improved only if any two terms can be improved. The first term cannot be improved by assumption. In case of second term the contribution cannot be improved from remaining asks. As earlier there are three possible cases here.

In case (1), there are no common set of asks which can be assigned to set of bids
Bd_k = \{B_1, \ B_2, \ldots, \ B_k\} and B_{k+1}. Each of this term is individually optimal. Hence following similar argument as in case of k = 2, it can be shown that the assignment is optimum.

In case (2), cov_{ak+1} can be improved by ask or set of asks currently assigned to bids BD_k=\{B_1, \ B_2, \ldots, \ B_k\}. If these asks are assigned to bid B_{k+1}, the ask or set of asks assigned to bid B_{k+1} can replace them without affecting any other assignment. There is no change in quantity replaced. So it is an interchange of asks hence will not change the value of objective function. It can be further proved that the second term cannot be improved following a similar procedure as done for k = 2.

In case (3), cov_{ak+1} can be improved by ask or set of asks currently assigned to bids in BD_k. However this requires assignment of currently unassigned asks to bids in BD_k. As earlier O_{nk} be the next highest contribution from assignment to bids in BD_k and cov_{nak+1}, is the maximum contribution from assignment to bid B_{k+1}. This cannot be improved by any other ask. As earlier in this case if

O_k + cov_{ak+1} > O_{nk} + cov_{nak+1}, the first assignment is optimum. In other cases second assignment is optimum. This follows by extending same argument as in earlier case, when k = 2. Hence the result follows by induction. So the algorithm generates optimum solution.

### 3.7 Payoff Determination Problem

Double Auctions can be used to implement efficient many to many electronic negotiations. There are two main problems in double auctions as follows.

1. Finding out optimal assignment of asks and bids. Algorithms for finding optimum assignments of asks and bids in case of assignment constraints and indivisibility constraints have been developed.

2. Designing a payment scheme for buyers and sellers such that auction mechanism satisfies properties like Incentive Compatible, Individual Rational, Budget Balance and Efficient. At this stage net payment by each buyer and net payment to each seller is to be computed. This will be referred to as Mechanism Design under assignment constraints.
Payment Mechanism: In the next step, the payment mechanism, which computes payoff of different buyers and sellers, is to be designed. Essential properties of any payment mechanism are:

1. **Strategy Proof (IC):** The payment mechanism should be strategy proof. It means that truthful bidding should be the dominant strategy. This property is also known as Incentive Compatibility.

2. **Individual Rationality (IR):** The payment mechanism should be individual rational. It means each buyer and seller should gain some amount by participating in auction. Otherwise there will not be any incentive for participating in auction.

3. **Budget Balance (BB):** The payment mechanism should be budget balanced. It means that difference between total payments to the sellers and total receipts from buyers should be nonnegative. It means that the auction should not be run in loss.

4. **Efficient (EF):** It means that the total profit obtained through the auction mechanism should be the maximum. The objects are assigned to those who value them the most.

However there is a well known impossibility result of Myerson and Satterthwaite [MS1983], which states that no mechanism can be efficient, budget balanced, incentive compatible and individual rational in respect of double auctions, at the same time. There can be many approaches to this problem.

**Approaches for Mechanism Design**

The payment mechanism, which is strategy proof, weakly budget balanced and individually rational is presented in [HWS2002]. This mechanism is based on single price for buyers and sellers. Other approach (known as VCG Mechanism) is to compute VCG Payments [V1961],[C1971],[G1973]. However VCG Payment mechanism is not budget balanced. This happens if amount to be paid to sellers exceeds the total receipts from buyers. In this case the auction has to run in loss. There can be number of approaches to achieve budget balance property.

1. One approach can be to design the payment scheme to minimize the distances from VCG Payments. This approach is followed in [PKE2001],[PKE2002].
(2) The participation fees from the buyer can be introduced and budget balance can be achieved. However in this case some buyers and sellers may not choose to participate. This can make the scheme inefficient. In our analysis we follow different payment schemes and compare them.

In VCG mechanism in the first step optimum assignment of asks and bids is obtained. Then amount payable by each buyer and receivable by each seller is computed. Let $amp_{bj}$ be the amount payable by $j^{th}$ buyer. Let $amp_{si}$ be the amount payable by $i^{th}$ seller. These amounts are computed as

$$amp_{bj} = bp_j - (V_o - V_{bj})$$  \hspace{1cm} (3.26)

where $V_o$ is the optimum solution of (3.1) or the total surplus from the optimum assignment of (3.1) and where $V_{bj}$ is the optimum solution of (3.1) or the total surplus from the optimum assignment of (3.1) after deleting $j^{th}$ buyer and keeping all other conditions same. In the same way the amount receivable by $i^{th}$ seller will be

$$amr_{si} = ap_i + V_o - V_{si}$$  \hspace{1cm} (3.27)

where $V_{si}$ is the optimum solution of (3.1) or the total surplus from the optimum assignment of (3.1) after deleting $i^{th}$ seller, keeping all other conditions same. So computation of VCG Payments requires that number of optimization problems should be solved.

### 3.8 VCG Payment Computation

In this section we derive some important results which can be used to VCG Payments without requiring solution of new optimization problems.

**Cyclic interchange of asks and bids:** Suppose that there is a set of bids $BD_k = \{B_i, i = 1, 2, ..., k\}$. Let $bq_i$ ($i = 1, 2, ..., k$) be the demand of $i^{th}$ bid. Let the total demand of these $k$ bids be $q = \sum_{i=1}^{k} bq_i$. Let $AS_{k1} = \{A_i, i = 1, 2, ..., k_1\}$ be the set of asks which is assigned to bid set $BD_k$. Let $aq_i$ ($i=1,2,..,k_1$) be the assigned quantity. Then we have $q = \sum_{i=1}^{k_1} aq_i$. The assignment is defined by set $A_{sg} = \{(B_i, A_j, aq_{ij}), i = 1, 2, ..., k, j = 1, 2, ..., k_1\}$.

So that quantity $aq_1$ of ask $A_1$ is assigned to bid $B_1$ and quantity $aq_{k1}$ of ask $A_k$ is assigned to bid $B_k$. Then we define cyclic interchange of quantity $q_c$ ($q_c \leq aq_1$) as a
change, which changes the assignment set $A_{sg}$ to \{(B_i, A_j, (aq_j - q_c)), i =1, 2, \ldots, k-1, j = 2, \ldots, k_1, (B_k, A_1, q_c)\}. A cyclic interchange of quantity $q_c$ in any assignment of asks to bids, will be called feasible if it is allowed as per the constraints. We call a feasible cyclic interchange as partial cyclic interchange, in case the assignment set gets changed as
\{(B_i, A_j, (aq_j - q_c)), i =1, 2, \ldots, k-1, j = 2, \ldots, k_1\}\)

In other words a cyclic interchange is an interchange where quantity $q_c$ of ask $A_1$ is assigned to bid $B_k$ and quantity $q_c$ of asks $A_i$ ($i= 2, 3, \ldots, k_1$) are assigned to bid $B_i$ ($i=2, 3, \ldots, k-1$). In case some asks $A_i$ ($i= 2, 3, \ldots, k_1$), have assigned quantity $aq_i < q_c$, then before interchanging asks are combined fully or partly such that assigned supply equals $q_c$. In partial cyclic interchange quantity $q_c$ of ask $A_1$ is not assigned to bid $B_k$ and the demand of quantity $q_c$ for bid $B_k$ remains unfulfilled.

Suppose that there are three bids, with bid prices $bp_1$, $bp_2$ and $bp_3$ respectively. Let $bq_1$, $bq_2$ and $bq_3$ be their respective quantities and $bs_1$, $bs_2$ and $bs_3$ be respective sizes. Suppose that optimum assignment of asks and bids, is as shown in Table 3.6.

<table>
<thead>
<tr>
<th>Bid</th>
<th>Bid Price</th>
<th>Ask</th>
<th>Ask Price</th>
<th>Ask Size</th>
<th>Bid Size</th>
<th>Quantity Assigned</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_1$</td>
<td>$bp_1$</td>
<td>$a_1$</td>
<td>$ap_1$</td>
<td>$as_1$</td>
<td>$bs_1$</td>
<td>$q_1$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$a_2$</td>
<td>$ap_2$</td>
<td>$as_2$</td>
<td></td>
<td>$q_2$</td>
</tr>
<tr>
<td>$b_2$</td>
<td>$bp_2$</td>
<td>$a_3$</td>
<td>$ap_3$</td>
<td>$as_3$</td>
<td>$bs_2$</td>
<td>$q_3$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$a_4$</td>
<td>$ap_4$</td>
<td>$as_4$</td>
<td></td>
<td>$q_4$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$a_5$</td>
<td>$ap_5$</td>
<td>$as_5$</td>
<td></td>
<td>$q_5$</td>
</tr>
<tr>
<td>$b_3$</td>
<td>$bp_3$</td>
<td>$a_6$</td>
<td>$ap_6$</td>
<td>$as_6$</td>
<td>$bs_3$</td>
<td>$q_6$</td>
</tr>
</tbody>
</table>

An example of cyclic interchange and partial cyclic interchange is shown in Table 3.7 and 3.8.

We state the following two results in case of bids. However they are symmetric and can be applied to asks as well.

**Theorem 8**: A feasible cyclic interchange of quantity $q_c$ in an optimum assignment will not change the value of objective function.
Proof: Suppose that in optimum assignment we have set of k bids $B_i$ with demand $b_{qi}$ ($i=1, 2, \ldots, k$). Suppose that set of $k_1$ asks $A_i$ ($i=1, 2, 3, \ldots, k_1$) with supply $a_{qi}$ be assigned to these set of k bids. So we have assignment set 
$$A_{sg} = \{(B_i, A_j, a_{qj}), i=1, 2, \ldots, k, j=1, 2, \ldots, k_1\}.$$ 
Suppose that a feasible cyclic interchange of quantity $q_c$ of ask $A_1$ is carried out. Then the assignment set will change to 
$$\{(B_i, A_j, (a_{qj} - q_c)), i=1, 2, \ldots, k, j=1, 2, \ldots, k_1\}.$$ 
In this interchange quantity $q_c$ of ask $A_1$ is assigned to bid $B_k$. In the same way quantity $q_c$ of asks $A_i$ ($i=2, 3, \ldots, k_1$) are assigned to bids $B_1, B_2, \ldots, B_{k-1}$. The quantity $q_c$ of ask $A_1$ is assigned to bid $B_k$. All these assignments are feasible. It can be seen that this interchange is equivalent to following set of interchanges.

(1) Interchange quantity $q_c$ of ask $A_1$ assigned to bid $B_1$ with quantity $q_c$ of ask $A_2$ assigned to bid $B_2$. This interchange does not change the value of objective function as well as surplus (By theorem 1).

(2) In next step interchange quantity $q_c$ of ask $A_1$ assigned to bid $B_2$ with quantity $q_c$ of ask $A_3$. This interchange does not change the value of objective function as well as surplus.

<table>
<thead>
<tr>
<th>Bid</th>
<th>Bid Price</th>
<th>Ask</th>
<th>Ask Price</th>
<th>Ask Size</th>
<th>Bid Size</th>
<th>Quantity Assigned</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_1$</td>
<td>$b_1 p_1$</td>
<td>$a_2$</td>
<td>$a p_2$</td>
<td>$a s_2$</td>
<td>$b s_1$</td>
<td>$q_2$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$a_4$</td>
<td>$a p_4$</td>
<td>$a s_4$</td>
<td></td>
<td>$q_1$</td>
</tr>
<tr>
<td>$b_2$</td>
<td>$b_2 p_2$</td>
<td>$a_3$</td>
<td>$a p_3$</td>
<td>$a s_3$</td>
<td>$b s_2$</td>
<td>$q_3$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$a_4$</td>
<td>$a p_4$</td>
<td>$a s_4$</td>
<td></td>
<td>$(q_4 - q_1)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$a_5$</td>
<td>$a p_5$</td>
<td>$a s_5$</td>
<td></td>
<td>$q_4$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$a_6$</td>
<td>$a p_6$</td>
<td>$a s_6$</td>
<td></td>
<td>$q_1$</td>
</tr>
<tr>
<td>$b_3$</td>
<td>$b_3 p_3$</td>
<td>$a_6$</td>
<td>$a p_6$</td>
<td>$a s_6$</td>
<td>$b s_3$</td>
<td>$q_6, q_1$</td>
</tr>
<tr>
<td>$b_3$</td>
<td>$b_3 p_3$</td>
<td>$a_1$</td>
<td>$a p_1$</td>
<td>$a s_6$</td>
<td>$b s_1$</td>
<td>$q_1$</td>
</tr>
</tbody>
</table>
Table. 3.8 Partial cyclic interchange

<table>
<thead>
<tr>
<th>Bid</th>
<th>Bid Price</th>
<th>Ask</th>
<th>Ask Price</th>
<th>Ask Size</th>
<th>Bid Price</th>
<th>Bid Size</th>
<th>Quantity Assigned</th>
</tr>
</thead>
<tbody>
<tr>
<td>b₁</td>
<td>bp₁</td>
<td>a₁</td>
<td>ap₁</td>
<td>as₁</td>
<td>bs₁</td>
<td>q₁</td>
<td></td>
</tr>
<tr>
<td>b₂</td>
<td>bp₂</td>
<td>a₂</td>
<td>ap₂</td>
<td>as₂</td>
<td>bs₂</td>
<td>q₂</td>
<td></td>
</tr>
<tr>
<td>b₃</td>
<td>bp₃</td>
<td>a₃</td>
<td>ap₃</td>
<td>as₃</td>
<td>bs₃</td>
<td>q₃</td>
<td></td>
</tr>
</tbody>
</table>

The above steps are repeated till bid Bₖ is reached and quantity qₖ is assigned to bid Bₖ. It may be noted that k is finite. As each interchange does not change the value of objective function and surplus the result follows.

**Theorem 9**: In case of partial cyclic interchange of quantity qₖ of ask A₁ with ask price ap₁ is not assigned to bid Bₖ with bid price bpₖ with qₖ being unfulfilled demand, then the value of objective function changes by

\[
\text{cov}_{1k}(q_c) = (bp_k - ap_1)q_c - (as_1 - bs_k)q_c .
\]

**Proof**: This result follows directly by application of earlier theorem. Let ov be the value of objective function in the optimum assignment.

Suppose that in optimum assignment we have set of k bids Bᵢ with demand bqᵢ (i=1, 2, …, k). Suppose that set of k asks Aᵢ (i=1, 2, 3,…k₁) with supply aqᵢᵢ be assigned to these set of k bids. So we have assignment set

\[ A_{sg} = \{(B_i, A_j, aq_j), i =1, 2, ..., k, j = 1, 2, ..., k₁\} \]

Suppose that a partial cyclic interchange of quantity qₖ of ask A₁ is carried out. Then the assignment set will change to

\[ \{(B_i, A_j, (aq_j . q_c)), i =1, 2, ..., k-1, j = 2, ..., k₁\} . \]

In case complete interchange of quantity qₖ is applied the value of objective function does not change. In this case the value of objective function is

\[ ov = ov_{k-1} + \text{cov}_{1k}(q_c) , \text{ov}_{k-1} \text{ is the contribution to the value of objective function, from assignment of bids excluding bid B}_k \text{ and ask A}_1. \text{ Let } \text{cov}_{1k}(q_c), \text{ be the contribution to the value of objective function by assigning ask A}_1 \text{ to bid B}_k. \text{ So the difference when complete interchange is carried out and partial interchange is carried out is} \]

\[ \text{cov}_{1k}(q_c) = (bp_k - ap_1)q_c - (as_1 - bs_k)q_c . \]
Further the change in surplus is \((bp_k - ap_1)q_c\).

These two results help in obtaining VCG payment without actually solving set of optimization problems. We start with an optimum assignment of (3.1). Suppose that there are \(n\) buyers. In order to obtain VCG payment of \(k^{th}\) buyer, we remove \(k^{th}\) buyer. After deleting \(k^{th}\) buyer we have an optimization problem with \((n-1)\) buyers and \(m\) sellers. Let \(BD_{k-1} = \{B_1, B_2, \ldots, B_{k-1}\}\), be the set bids of buyers 1, 2, \ldots, \(k-1\). Similarly let \(BD_2\), be set of bids of buyers \((k+1)\) to \(n\). Let \(AS_1\) and \(AS_2\) be the set of asks assigned to bid sets \(BD_{k-1}\), prior to assignment to bid \(B_k\), and \(BD_2\). Let \(AS_k\) be the set of asks assigned to \(k^{th}\) buyer’s bid.

\[
\text{Algorithm VCG}(BD, AS, k) \{
\quad \text{while ( new buyer) } \{
\quad \text{ for ( k + 1 to n ) do } \{
\quad \quad \text{ add current ask to unassigned ask set }
\quad \quad \text{ for current bid determine the ask with the highest unit contribution ; }
\quad \quad \text{ assign ask }
\quad \} \quad \text{ change in surplus } = (\text{current_bid_price} - \text{unassigned_ask_price}) \text{ current bid}
\text{ while ( new seller) } \{
\quad \text{ for ( k + 1 to m ) do } \{
\quad \quad \text{ add current bid to unassigned bid set }
\quad \quad \text{ for current ask determine the bid with the highest unit contribution ; }
\quad \quad \text{ assign bid}
\quad \} \quad \text{ change in surplus } = (\text{current_bid_price} - \text{unassigned_ask_price}) \text{ current bid}
\}\}
\]

Figure. 3.13 – Algorithm for computation of VCG Payment

In order to determine VCG payment of \(k^{th}\) buyer, we first determine the set of bids, to which the asks in \(AS_k\) (assigned to the bid of \(k^{th}\) buyer) can be assigned. In this case it is not necessary to verify the bids in set \(BD_{k-1}\). This is due to the fact our optimization algorithm assigns asks with the highest unit contribution to any bid at any point of time. So all the asks in set \(AS_1\), which are assigned to bids in set \(BD_{k-1}\) have higher unit contribution than the asks in set \(AS_k\). It is not necessary to carry out assignment again, as we have optimum assignment for \(BD_{K-1}\). So we start assignment from \((k+1)^{th}\) bid and continue verifying bids in set \(BD_2\). We add asks in set \(AS_k\) to set \(AS_2\). Let \(AS_3\) be this set.
Then the assignment between \(BD_{k-1}\), \(BD_2\) and set of asks in \(AS_3\) is carried out by applying our optimization algorithm. This process is continued till all bids in set \(BD_2\) are assigned. As the supply now is more than demand, quantity equal to demand of \(k^{th}\) buyer from some ask will remain unfulfilled. So the surplus from this ask and \(k^{th}\) buyer ask will give the change in the value of surplus and then using (3.26) and (3.27) VCG payment can be obtained. This process is repeated for all buyers and sellers. Then we obtain VCG payment for each buyer and seller in polynomial time.

Let \(BD\) be the set of bids and \(AS\) be the set of asks. So the main algorithm to compute VCG payment is shown in figure 3.13.

**Theorem 10**: The solution obtained by above algorithm is optimum for new problem of \((n-1)\) buyers and \(m\) sellers and vice-versa.

**Proof**: Suppose that there are \(n\) buyers and \(m\) sellers in original optimization problem. In order to obtain VCG payment of \(k^{th}\) buyer, we remove \(k^{th}\) buyer. After deleting \(k^{th}\) buyer we have an optimization problem with \((n-1)\) buyers and \(m\) sellers. Let \(BD_{k-1} = \{B_1, B_2, \ldots, B_{k-1}\}\), be the set bids of buyers 1, 2, \ldots, \(k-1\). Similarly let \(BD_2\), be set of bids of buyers \((k+1)\) to \(n\). Let \(AS_k\) and \(AS_2\) be the set of asks assigned to bid sets \(BD_{k-1}\) and \(BD_2\) respectively. Let \(AS_k\) be the set of asks assigned to \(k^{th}\) buyer’s bid. Let \(AS_3\) be the set of asks such that \(AS_3 = AS_2 \cup AS_k\).

The set of asks in \(AS_1\) are assigned to the set of bids in set \(BD_{k-1}\). Let \(ov_1\) be the contribution to value of objective function from this assignment. In our algorithm the assignment is the set of asks in \(AS_3\) is carried out with set of bids in \(BD_{k-1}\) and \(BD_2\). After this assignment we get an assignment which optimum for bids in set \(BD_{k-1}\) and \(B_{k+1}\). This process is carried out till assignment of all remaining bids is completed.

At each stage we get optimum assignment. It may be noted that number of buyers is finite. So the assignment is the optimum and the surplus generated is also the optimum surplus. Since there is no assumption about value of \(k\), the result is true for all buyers.

This argument can be easily extended for group of asks or in case of sellers.

It can be seen that the time complexity of the algorithm is always polynomial. It can be seen that an ask or bid with highest unit contribution can be determined by scanning \((n-1)\) asks or \((m-1)\) bids in worst case for a single bid or ask. There are \((n-1)\) asks and \(m\) bids
or vice versa. So an optimum solution can be obtained with worst case time complexity of \( m(n-1) \) or \( n(m-1) \). In worst case scenario all VCG payment for all buyers and sellers can be obtained with worst time complexity of \( O(n^3) \). It can be seen that in case of optimization formulation (1), the performance of this algorithm will be linear, as the change in surplus can be determined by subtraction of last bid and first ask in the set. If a table maintaining unit contribution of different asks and bids is created then the performance of the algorithm can be speeded up considerably.

### 3.9 Achieving Budget Balance

VCG Payment mechanism is not budget balanced. This means that the total amount payable to all sellers exceeds the total amount receivable from all buyers. In such a scenario auction has to run in loss.

One approach to achieve budget balance has been presented in [PKE2001]. In this the total surplus is allocated to individual buyers in such a way that the distance to Vickrey discount of each individual buyers and sellers is minimized. The problem of distance minimization is formulated as constrained optimization problem treating budget balance (BB) and individual rationality (IR) as hard constraints. However, this approach is not truthful i.e. truthful bidding is not the dominant strategy.

Let \( V \) be the total surplus obtained after optimum assignment. Suppose that VCG Discount for each participant has been obtained. A participant in the transaction can be a buyer or a seller. This discount is obtained for individual buyers and sellers. Let there be \( n \) buyers and \( m \) sellers. In all there are \( (n + m) \) participants. Let \( bv_i \) and \( sv_j \) be the Vickrey discounts of \( i^{th} \) buyer and \( j^{th} \) seller respectively. Let \( bd_i \) and \( sd_j \) be the discounts of \( i^{th} \) buyer and \( j^{th} \) seller, so that the scheme becomes budget balanced. These discounts are computed in such a way, they minimize the distance from VCG discounts.

Let \( D(bd_i, sd_j, bv_i, sv_j) \) be distance function. Then the problem is formulated as

\[
\begin{align*}
\text{min } & D(bd_i, sd_j, bv_i, sv_j) \\
\text{s.t. } & \Sigma bd_i + \Sigma sd_j \leq V \\
& bd_i \leq bv_i \text{ for } i = 1, 2, \ldots, n \\
& sd_j \leq sv_j \text{ for } j = 1, 2, \ldots, m \\
& bd_i \geq 0 \text{ for } i = 1, 2, \ldots, n
\end{align*}
\]
\[ \text{sd}_j \geq 0 \quad \text{for } j = 1, 2, \ldots, m \quad (3.33) \]

In this case constraint in (3.29) ensures budget balance. The constraints in (3.30) and (3.31) ensure that the individual discount of each buyer and seller does not exceed their respective Vickrey discount. The constraints (3.32) and (3.33) ensure individual rationality. So all the participating buyers and sellers make positive gains. The different types of distance function considered are

1. square distance i.e. \( \sum (\text{bv}_i - \text{bd}_i)^2 + \sum (\text{sv}_j - \text{sd}_j)^2 \),
2. maximum distance
3. relative error function
4. squared relative error
5. weighted error function.

Then the problem is not directly solved, an analytic function for the solutions that correspond to each distance function is obtained. Each family of solution is a parameterized payment rule. The buyers and sellers (participants) are arranged in the order of Vickrey discounts. The different payment rules those are implemented are as follows:

Threshold: If threshold parameter falls into interval k, then first k participants receive discount, while others do not receive any discount.

Small: If parameter falls into interval k, then the first k participants do not receive any discount, while others receive.

Fractional: In this scheme each participant receive discount, which is equal to proportional share of Vickrey discount of the participant.

Large: This is reverse of small.

Reverse: This is reverse of threshold.

Pay what you bid: In this scheme the buyer pays the amount, which is equal to product of his bid price and the quantity assigned. The seller gets the amount equal to the product of his ask price and the quantity assigned.
VCG Discount: In this scheme, VCG discounts are computed and each buyer and seller gets VCG Discount. So the amount payable by the buyer will be the difference between product of bid price and quantity assigned, and VCG Discount of the buyer. Each seller will get an amount equal to the sum of the ask price and the quantity assigned, and VCG Discount of the seller.

It can be seen that in above rules, except in case of fractional rule, there is possibility of some buyers and sellers not getting any discount. If VCG scheme is not budget balanced then our scheme computes the pay off so that budget balance is achieved.

In this section, we work out the payment scheme which is budget balanced as follows: The payment by each buyer and to be paid to each seller has to be determined after the optimum assignment has been found out. The main problem is how to distribute the total surplus obtained. The budget balance constraint means that total payment from all the buyers and the total amount to be paid to all the sellers must be the same. It is also desirable that electronic exchange remains neutral between buyers and sellers. In double auctions with assignment constraints, different buyers and sellers trade multiple units of goods, on multiple attributes, with aggregation over demand and supply. In such a scenario it may not be possible to decide the criteria on distribution of surplus. We attempt to determine some criteria for distribution of surplus.

3.10 Payment Scheme Based on Contribution
Let $BD_i$ be the set of bids selected from $i^{th}$ buyer in the optimum assignment. It consists of three elements viz., bid price ($bp_i$), matching ask price ($ap_k$) and quantity matched $q_i$. Then we define the contribution of the $i^{th}$ buyer to the total surplus as

$$covs_{bi}(q) = \sum_{BD_i} (bp_i - ap_j)q_j$$ (3.34).

The summation is over all elements of set $BD_i$. If $\sum_{BD_i} q_i = q$, it means that the demand of $i^{th}$ buyer is completely fulfilled, otherwise it will be partially fulfilled. In case the demand of $i^{th}$ buyer has a single bid with indivisibility constraint, there will be only one element in set $BD_i$. The buyer’s contribution is sum of products of differences between matched bids and asks and quantity matched in between different asks and bids. It can be easily seen that the total
surplus from matching will be sum of all the contributions of all buyers whose bids are matched. So we get
\[ V = \sum_{BD} \text{covs}_{b}(q) \]  
(3.35)
where \( BY \) is the set of buyers, whose bids are partly or fully matched. Let \( AS_j \) be the set of asks selected from \( j^{th} \) seller in the optimum assignment. It consists of three elements ask price (\( ap_j \)), matching bid price (\( bp_i \)) and quantity matched \( q_j \). Then we define the contribution of the \( i^{th} \) seller to the total surplus as
\[ \text{covs}_{As_j}(q) = \sum_{AS_j} (bp_i - ap_j)q_j \]  
(3.36).
The summation is over all elements of set \( AS_j \). (Its definition is symmetric to that of buyer’s contribution).
If \( \sum_{AS_j} q_j = q \), it means that the supply of \( j^{th} \) seller is completely utilized, otherwise it will be partially utilized. In case the ask of \( j^{th} \) seller is completely assigned to a bid with indivisibility constraints, there will be only one element in set \( AS_j \). The seller’s contribution is the sum of products of differences between matched bids and asks and quantity matched in between different asks and bids. It can be easily seen that the total surplus from matching will be sum of all the contributions of all sellers whose bids are matched. So we get
\[ V = \sum_{SL} \text{covs}_{s}(q) \]  
(3.37).
where \( SL \) is the set of sellers, whose bids are partly or fully matched.

The contribution is used as the value that each buyer and seller brings to the assignment problem. This concept is similar to the concept of added value concept in game theory [BS1996]. Added value measures how much each player contributes to the game. It is computed as the difference between the surplus when all participants are there and the surplus by removing one participant each time. It is computed for each participant. However instead of added value, we use buyer’s and seller’s contribution The reason for using this is that in case of double auctions with assignment constraints, it may not be possible to get added value, as removing a seller means it may not be possible to match certain bids. This will make computation of added value difficult. So we use each participant’s contribution in the same way as added value. We can define certain fairness
criteria for surplus distribution. The surplus can then be distributed using these criteria as follows.

(1) A participant gets higher proportion of surplus if his contribution is higher. If $i^{th}$ buyer has higher contribution than $(i+1)^{th}$ buyer, then $i^{th}$ buyer gets higher proportion of surplus. The fairness of the scheme also means that a participant with no contribution, does not get any proportion of surplus and the one with the same contribution gets the same proportion of surplus.

(2) A proportion of surplus should be linear function of contribution and should vary at constant rate with change in contribution. This makes that payments worked out are fair to all the participants, and there is no favour to any particular buyer or seller.

The budget balance constraint means that the sum of proportions of surplus for both the buyers and the sellers should be 1. In this case we allocate exactly half of the available surplus to buyers and remaining half to the sellers. Let $sp_b_i$ and $sp_s_j$ be the proportion of surplus allocated to $i^{th}$ buyer and $j^{th}$ seller respectively. The proportion for the $i^{th}$ buyer can be calculated as follows

$$sp_b_i = \frac{2covs_b_i(q)}{V}$$  

(3.38)

It can be seen that the payment scheme worked out satisfies the requirements stated in (1) and (2).

$$sp_b_i = sl(covs_b_i(q)) + inc,$$  

where $sl$ and $inc$ are constants.

This follows from the linearity assumption and the requirement that if $covs_b_i(q)$ is same then the proportion should be same. Since if contribution is 0 the proportion is also 0.

This means that $inc = 0$ and we get

$$sp_b_i = sl(covs_b_i(q))$$  

(3.39).

Adding it for all buyers we get

$$\Sigma sp_b_i = \Sigma sl(covs_b_i(q))$$  

(3.40)

Since $\Sigma sp_b_i = 1$, we get

$$sl \Sigma covs_b_i(q) = 1 \quad sl = \frac{1}{V}$$  

(3.41)
This means \( sp_{bi} = \frac{SP_{bi}}{V} \).

Apart from payment based on surplus the payment schemes based on quantity, price-quantity product and VCG discount are also proposed.

**Payment Scheme based on Quantity**

We can also use quantity purchased for distribution of surplus, instead of contribution. This can also be considered as a fair scheme, since the surplus is distributed as per quantity purchased. Extending the above argument to quantity purchased means that:

1. A participant gets higher proportion of surplus if quantity purchased is higher. If \( i^{th} \) buyer purchases higher quantity than \( (i+1)^{th} \) buyer, then \( i^{th} \) buyer gets higher proportion of surplus. The fairness of the scheme also means that a participant with no quantity sold or purchased does not get any proportion of surplus and the one with same quantity gets the same proportion of surplus.

2. A proportion of surplus should be linear function of quantity purchased/sold and should vary at constant rate with change in contribution. This makes that payments worked out is fair to all the participants, and there is no favour to any particular buyer or seller.

In the same way as above we can work out the formula for proportion as

\[ sp_{bi} = \frac{q_{bi}}{q}, \] where \( q \) is the total quantity and \( q_{bi} \) is the quantity purchased by \( i^{th} \) buyer. In this scheme some buyers or sellers may not gain much by bidding untruthfully. It can be seen that the buyer or seller purchasing smaller amount of quantity can make only limited gain, though the scheme may not prevent them from bidding untruthfully. We do not consider the case of misreporting of other attributes by buyers here. The requirements of buyers change (e.g. different widths), if they misreport other attributes.

**Payment Scheme based on Price Quantity Factor**

It is expected that buyer who bids with higher price and higher quantity need not have higher contribution than a buyer with lower price. This can happen in case of constrained
assignment, where buyer’s bid can be matched with an ask of higher price (though lower than matching bid price). So the combination of price and quantity (i.e. pay what you bid can also be used for surplus) is used. So the proportion can also be calculated as

\[ sp_{bi} = \frac{p_{bi}}{\sum p_{bi}} \]  

(3.42).

where \( pb_i \) is the amount payable by \( i^{th} \) buyer in case of no discount.

**Payment Scheme based on Discount**

Another method that can be used, for calculating discount can be based on Vickrey discount for individual participant. We can use following two functions for calculating the Vickrey discount.

\[ df = \frac{V}{\sum_i bdi + \sum_j sdj} \]

It will be a constant factor for all buyers and sellers.

The other formula can be to multiply individual discount by \( \frac{1}{\sum_i bdi + \sum_j sdj} \) to arrive at \( df_{bi} \), the individual discount factor for \( i^{th} \) buyer. Let \( bd_i \) be the discount allocated to the \( i^{th} \) buyer by any of the above schemes. Then the amount payable of \( i^{th} \) buyer is

\[ bpa_i = bp_i q_i - bd_i \]  

(3.43)

Where \( bp_i \) is the per unit bid price of the bid submitted by \( i^{th} \) buyer and \( q_i \) is the quantity allocated. Since \( bd_i > 0 \), each buyer pays less than what he bids, and hence makes positive gain. So the scheme is individual rational (IR). The scheme is also budget balanced (BB), since the total amount payable to the sellers is equal to the total amount receivable from the buyers. So the auction does not run in loss. The scheme is efficient as the objects are assigned to those who value them the most. Further it can be seen that \( bd_i \leq V \) for all \( i \), so the maximum gain by any buyer is always bounded. The same holds for the seller. In the next section the effect of changing prices by different buyers is studied. It may be noted that that all the schemes are efficient. The analysis has been carried out in respect of payment scheme by contribution. The analysis in case of other payment schemes is similar and hence it is omitted. We carry out the experimental analysis for all the above schemes.
3.11 Effect of Changing Bid Price by a Single Buyer

The above schemes are not strategy proof. So truthful bidding is not the dominant strategy. So we attempt to find the gain that buyer can make by bidding untruthfully and how the effect can be minimized. Suppose that each buyer and seller have their private valuation and they use it while bidding. Initially it is assumed that only one buyer bids untruthfully, while others bid truthfully. Let \( bp_v \) be the private valuation of the \( i \)th buyer. Let \( bpa_i \) be the amount payable by \( i \)th buyer when auction clears. Then gain (also called as utility of the buyer) of the buyer \( bg_i \) is

\[
bg_i = bpa_i - bp_v \tag{3.44}
\]

It can be seen that the buyer can improve his gain by bidding for a lower amount, by decreasing the quantity required and keeping price unchanged or by decreasing both. However without loss of generality it can be assumed that the buyer prefers to lower price rather than lowering the quantity required. Also price and quantity are generally related e.g. buyer is willing to pay lower price for higher quantity and so on. Hence it is assumed that the buyer is willing to lower price rather than quantity, while bidding. In bidding truthfully the \( i \)th buyer submits a bid with bid price based on his true valuation of the payable amount. Let \( bp_i \) be the bid price submitted by the buyer based on his true valuation. However if truthful bidding is not the dominant strategy, the buyer sets his bid price as

\[
butp_i = bp_i - x \tag{3.45}
\]

where \( x \) is a positive real number.

If the \( i \)th buyer bids the above amount, whereas all other bid truthfully the surplus will change by amount \( xq_i \), where \( q_i \) is the quantity allocated. The surplus decreases because the buyer submits lower price. So the new surplus will be

\[
V_n = V - xq_i \tag{3.46}
\]

The second term is positive and hence the \( V_n < V \).

The \( i \)th buyer’s contribution decreases by \( xq_i \). There will also be similar decrease in price quantity multiplication factor.

**Theorem 11**: If only one buyer decreases the amount
(1) the proportion of surplus allocated to that buyer also decreases under contribution payment scheme 
(2) the scheme allocates entire decrease in surplus to the buyer. The utilities of all others remain unchanged and the gain of the buyer (who decreases his price) is always bounded.

Proof: It can be seen that if \( i^{th} \) buyer decreases his price by \( x \), the contribution and the surplus decreases by \( xq_i \).

Let \( V \) be the total surplus. Let \( \text{covsb}_i(q) \) be the contribution of \( i^{th} \) buyer to the surplus. Let \( \text{ocovs} \) be the contribution of other buyers. Then we can write

\[
V = \text{covsb}_i(q) + \text{ocovs} \quad (3.47)
\]

If the \( i^{th} \) buyer decrease his bid price by \( x \), the contribution will change to

\[
\text{covsn}_i(q) = \text{covsb}_i(q) - xq_i \quad (3.48).
\]

The surplus will change to

\[
V_n = V - xq_i = \text{covsb}_i(q) + \text{ocovs} - xq_i
\]

\[
= ( \text{covsb}_i(q) - xq_i ) + \text{ocovs} \quad (3.49)
\]

So the proportion of surplus that is allocated changes in two schemes as in contribution based scheme

\[
\text{spn}_{bi} = \frac{2\text{cov sn}_i(q)}{V_n} = \frac{2\text{covsb}_i(q) - 2xq_i}{( \text{covsb}(q) - xq_i ) + \text{ocovs}} \quad (3.50)
\]

If the buyer bids truthfully then the proportion will be

\[
\text{sp}_{bi} = \frac{2\text{covsb}_i(q)}{V} = \frac{2\text{covsb}(q)}{\text{covsb}(q) + \text{ocovs}} \quad (3.51)
\]

Substituting \( a = \text{covsb}_i(q) \), \( b = \text{ocovs} \), \( y = xq_i \), in (3.50) and (3.51) these expressions can be written as

\[
\text{spn}_{bi} = \frac{2a - 2y}{(a - y + b)} \quad (3.52)
\]
\[
sp_{bi} = \frac{2a}{(a + b)}
\]

Subtracting (3.52) from (3.53) we get
\[
sp_{bi} - sp_{ni} = \frac{2a}{(a + b)} - \frac{2a - 2y}{(a - y + b)}
= \frac{2a(a - y + b) - (a + b)(2a - 2y)}{(a + b)(a - y + b)}
= \frac{2a a - 2ay + 2ab - 2aa + 2ay - 2ab + 2by}{(a + b)(a - y + b)}
= \frac{2by}{(a + b)(a - y + b)}
\]

(3.54)

Since \(y > 0\) and \(y < a\), the above term is always positive. So
\[
sp_{bi} - sp_{ni} > 0
\]

So the proportion of surplus allocated to the \(i^{th}\) buyer decreases. The discount also decreases (As there is decrease in surplus).

The discount of \(i^{th}\) buyer \(db_i\), when he bids truthfully is
\[
db_i = sp_{bi} V = \left( \frac{2covsb(q)}{V} \right) V = 2covsb_i(q)
\]

The discount of \(i^{th}\) buyer \(dbn_i\), when he bids untruthfully is
\[
dbn_i = sp_{ni} V = \left( \frac{2covsnb(q)}{V} \right) V_n = 2covsnb_i(q) = 2(covsb_i(q) - x_qi)
\]

So change in discount is \(db_i - dbn_i = 2x_qi > 0\). So entire decrease in surplus is allocated to \(i^{th}\) buyer.

So the difference in amount payable is
\[
bp_{i} q_{i} - sp_{bi} V - (bp_{i} q_{i} - x_qi - sp_{ni} V_n)
= bp_{i} q_{i} - 2covsb_i(q) - (bp_{i} q_{i} - x_qi - 2(covsb_i(q) - x_qi))
= bp_{i} q_{i} - 2covsb_i(q) - (bp_{i} q_{i} - x_qi - 2covsb_i(q) + 2 x_qi)
= bp_{i} q_{i} - 2covsb_i(q) - (bp_{i} q_{i} + x_qi - 2covsb_i(q))
= bp_{i} q_{i} - 2covsb_i(q) -bp_{i} q_{i} - x_qi + 2covsb_i(q) = - x_qi.
\]

So \(i^{th}\) buyer can improve utility only by \(x_qi\). The scheme does not allocate any additional discount. This can be similarly proved for price quantity scheme.
Further if only $i^{th}$ buyer decrease bid price and prices of others remain the same, then the proportion of discount of $j^{th}$ buyer ($j \neq i$) increases. If all buyers bid truthfully, then the proportion of surplus to be allocated to $j^{th}$ buyer is

$$sp_{bj} = \frac{2\text{covsb}_j(q)}{V} ,$$

where $\text{covsb}_j(q)$ be the contribution of $j^{th}$ buyer to the surplus.

If $i^{th}$ buyer decreases, his bid price and no other buyer change their respective bid prices, then the proportion of surplus to be allocated to $j^{th}$ buyer is

$$sp_{n_{bj}} = \frac{2\text{covsb}_b(q)}{V_n} ,$$

as $V > V_n$, we get $sp_{bj} < sp_{n_{bj}}$. So discount of all other buyers improve. So the new discount of $j^{th}$ buyer is

$$db_{n_j} = sp_{n_{bj}} V_n = \left(\frac{2\text{covsb}_b(q)}{V_n}\right)V_n = \text{covsb}_j(q) = db_j$$ (3.55).

As there is no change in bid price and discount, the utilities of others do not change. So if any buyer decreases his bid price, his gain is always be bounded and the utilities or the payable amount of other buyers remain unchanged. The gain is bounded. This is due to the fact that if the buyer continues to decrease his price, he may be out of auction after some stage. The scheme is not strategy proof, however it ensures that the gain of buyer will be bounded and that of others who bid truthfully remains does not change. This analysis can similarly be extended to seller.

Extending this result to the scenario where more than one buyer or seller bid untruthfully may not be straightforward. If two or more buyers bid untruthfully, the proportion of surplus allocated to all may not decrease. This can be seen from the following.

Suppose that two buyers $i$ and $j$ bid untruthfully. As earlier let $\text{covs}_{bi}(q)$ and $\text{covs}_{bj}(q)$ be their respective contributions to the surplus $V$ when bidding truthfully. Let $ocovs$ be the contribution of other buyers. Suppose that by bidding untruthfully the contribution decrease by $x$ and $y$ respectively. The proportion of surplus that is allocated to these two buyers is

$$sp_{bj} = \frac{a}{(a + b + c)} ,$$

where $a = \text{covs}_{bi}(q)$, $b = \text{covs}_{bj}(q)$ and $c = ocovs$. 

119
\[ sp_{bij} = \frac{b}{(a + b + c)} \]

The proportion of discount, when they bid untruthfully is

\[ spn_{bij} = \frac{(a - x)}{((a - x) + (b - y) + c)} \quad (3.56) \]

\[ spn_{bij} = \frac{(b - y)}{((a - x) + (b - y) + c)} \quad (3.57) \]

The proportion that is allocated to buyer B, will change by

\[
\begin{align*}
\frac{a}{(a + b + c)} - \frac{(a - x)}{((a - x) + (b - y) + c)} &= \frac{a((a - x) + (b - y) + c) - (a - x)(a + b + c)}{(a + b + c)((a - x) + (b - y) + c)} \\
&= \frac{(aa - ax + ab - ay + ac) - aa - ab - ac + ax + xb + xc}{(a + b + c)((a - x) + (b - y) + c)} \\
&= \frac{bx + cx - ay}{(a + b + c)((a - x) + (b - y) + c)}
\end{align*}
\]

This expression is positive if \((bx+cx-ay) > 0\). In this case the proportion increases. It decreases if \((bx+cx-ay) < 0\), and does not change if \((bx+cx-ay) = 0\).

In the same way the difference in proportion of the second buyer is

\[
\begin{align*}
\frac{b}{(a + b + c)} - \frac{(b - y)}{((a - x) + (b - y) + c)} &= \frac{b((a - x) + (b - y) + c)}{(a + b + c)((a - x) + (b - y) + c)} \\
&= \frac{(ab - bx + bb - by + bc) - ab - bb - bc + ay + yb + yc}{(a + b + c)((a - x) + (b - y) + c)} \\
&= \frac{ay + cy - bx}{(a + b + c)((a - x) + (b - y) + c)}
\end{align*}
\]

This expression is positive if \((ay+cy-bx)>0\). In this case proportion increases.

It decreases if \((ay+cy-bx)<0\), and does not change if \((ay+cy-bx) = 0\). However this does not guarantee the discount increase as the overall surplus also goes down. We can generalize this as follows.
Suppose that three buyers i, j and k bid untruthfully. As earlier let \( \text{covs}_b(q) \), \( \text{covs}_j(q) \) and \( \text{covs}_k(q) \) be their respective contributions to the surplus V when bidding truthfully. Let \( \text{o covs} \) be the contribution of other buyers. Suppose that by bidding untruthfully the contribution decrease by \( x \), \( y \) and \( z \) respectively. The proportion of surplus that is allocated to these three buyers is

\[
\text{sp}_{bi} = \frac{a}{(a + b + c + d)},
\]

where \( a = \text{covs}_b(q) \), \( b = \text{covs}_j(q) \) and \( c = \text{covs}_k(q) \), \( d = \text{o covs} \).

\[
\text{sp}_{bj} = \frac{b}{(a + b + c + d)}
\]

\[
\text{sp}_{bk} = \frac{c}{(a + b + c + d)}
\]

The proportion of discount, when they bid untruthfully is

\[
\text{spn}_{bi} = \frac{a - x}{((a - x) + (b - y) + (c - z) + d)}
\]

\[
\text{spn}_{bj} = \frac{b - y}{((a - x) + (b - y) + (c - z) + d)}
\]

\[
\text{spn}_{bk} = \frac{c - z}{((a - x) + (b - y) + (c - z) + d)}
\]

The proportion that is allocated to buyer i will change by

\[
\frac{a}{(a + b + c + d)} - \frac{a - x}{((a - x) + (b - y) + (c - z) + d)}
\]

\[
= \frac{a((a - x) + (b - y) + (c - z) + d) - (a - x)(a + b + c + d)}{(a + b + c + d)((a - x) + (b - y) + (c - z) + d)}
\]

\[
= \frac{(aa - ax + ab - ay + ac - az + ad) - aa - ab - ac - ad + ax + xb + xc + xd}{(a + b + cd)((a - x) + (b - y) + (c - z) + d)}
\]

\[
= \frac{bx + cx + dx - ay - az}{(a + b + cd)((a - x) + (b - y) + (c - z) + d)}
\]

This expression is positive if \( bx + cx + dx - ay - az > 0 \). In this case the proportion increases. It decreases if \( (bx + cx + dx - ay - az) < 0 \), and does not change if \( (bx + cx + dx - ay - az) = 0 \). In the same way the difference in proportion of the second buyer j is
\[
\frac{b}{(a + b + c + d)} - \frac{b - y}{((a - x) + (b - y) + (c - z) + d)},
\]

which simplifies to

\[
= \frac{ay + cy + dy - bx - bz}{(a + b + c + d)((a - x) + (b - y) + (c - z) + d)}
\]

This is positive if \((ay+cy+dy-bx-bz)>0\). In this case the proportion increases. It decreases if \((ay+cy+dy-bx-bz)<0\), and does not change if \((ay+cy+dy-bx-bz)=0\). However this does not guarantee that the discount increases as overall surplus goes down.

In the same way the difference in proportion of the third buyer \(k\) is

\[
\frac{c}{(a + b + c + d)} - \frac{c - z}{((a - x) + (b - y) + (c - z) + d)}
\]

Which can be simplified as

\[
= \frac{ay + by + dy - cx - cy}{(a + b + c + d)((a - x) + (b - y) + (c - z) + d)}
\]

This expression is positive if \((ay+by+dy-cx-cy)>0\). In this case proportion increases. It decreases if \((ay+by+dy-cx-cy)<0\), and does not change if \((ay+by+dy-cx-cy)=0\). However this does not guarantee that the discount increases as overall surplus goes down.

Generalizing this for \(k\) buyers, we obtain the general formula for change in the proportion of surplus that can be allocated. The change will be given as under.

Let \(ccov_{bi}\) be the change in contribution of \(i\)th buyer contribution by bidding untruthfully.

Then change in proportion is positive, negative or 0 depending upon

\[
ccov_{bi} \sum_{j=1}^{k} cov_{b_j}(q) - cov_{b_i}(q) \sum_{j=1}^{k} cccov_{bi}
\]

The new discount is

\[
df_{bi} = \frac{(cov_{b_i}(q) - cccov_{bi})}{V - \sum_{i} cov_{b_i}}\left(V - \sum_{i} cccov_{bi}\right)
\]

This gives idea about the effect of untruthful bidding by any buyer. However our contribution payment scheme ensures that even if any \(k\) of \(n\) buyers decrease their respective bid prices, the utilities of others remain does not change. The decrease in surplus is allocated among the \(k\) buyers only irrespective of changes in proportion. We state this as
Theorem 12: If any k buyers decreases the amount

(3) the decrease in surplus is allocated to these k buyers under contribution payment scheme.

(4) the utilities of all others remain unchanged and the gain of the k buyers (who decrease their prices) is always bounded.

Proof: Let V be the total surplus. Let covsb_i(q) be the contribution of i\(^{th}\) buyer to the surplus. It is assumed without any loss of generality that buyers 1, 2, ..., k decrease their respective bid prices by x_i. Let q_i be the quantities purchased by them. The buyers k+1, k+2, ..., n do not change their respective bid prices. Let V_n be the new surplus. Let spbi be the proportion of surplus allocated, when all the buyers bid truthfully. Let spnbi be the new proportion of surplus allocated, when k buyers bid decrease their respective bid prices. It can be seen that

\[ V_n = V - \sum_{i=1}^{k} x_i q_i \]

For buyers i = 1, 2, ..., k the new contribution will be

\[ \text{covsnb}_i(q) = \text{covsb}_i(q) - x_i q_i \] . The contributions of others remain unchanged.

The discount of i\(^{th}\) buyer db_i, when he bids truthfully is (i = 1, 2, ..., k)

\[ db_i = spbi \cdot V = \left( \frac{2\text{covsb}_i(q)}{V} \right) V = 2\text{covsb}_i(q) . \]

The discount of i\(^{th}\) buyer dbn_i, when he bids untruthfully is

\[ dbn_i = spnbi \cdot V = \left( \frac{2\text{covsnbi}(q)}{V} \right) V_n = 2\text{covsnb}_i(q) = 2(\text{covsb}_i(q) - x_i q_i) . \]

So change in discount is db_i - dbn_i = 2x_i q_i > 0. In the same way as earlier it can be shown that the change in amount payable is x_i q_i. So i\(^{th}\) buyer can improve utility only by x_i q_i. The scheme does not allocate any additional discount.

Further if buyers (i = 1, 2, ..., k) decrease bid their respective prices and prices of others remain the same, then the proportions of discount to remaining buyers increase. If all buyers bid truthfully, then the proportion of surplus to be allocated to j\(^{th}\) (j > k) buyer is

\[ spbj = \frac{2\text{covsb}_j(q)}{V} , \] where \( \text{covsb}_j(q) \) be the contribution of j\(^{th}\) buyer to the surplus.
If buyers \((i = 1, 2, \ldots, k)\) decrease their respective bid prices and no other buyers change their respective bid prices, then the proportion of surplus to be allocated to \(j^{th}\) \((j > k)\) buyer is

\[ sp_{nbj} = \frac{2covsb(q)}{V_n}, \]

as \(V > V_n\), we get \( sp_{bj} < sp_{nbj} \). So discount of all other buyers improve. So the new discount of \(j^{th}\) \((j>k)\) buyer is

\[ dbn_j = sp_{nbj} V_n = \left(\frac{2covsb(q)}{V_n}\right)V_n = covsb_j(q) = db_j \]  \hspace{1cm} (3.61).

As there is no change in bid price and discount, the utilities of others do not change. So even if any \(k\) buyers decrease their respective bid prices, their gains will always be bounded and the utilities or the payable amount of other buyers remain unchanged. The gain is always bounded. This is due to the fact that if the buyers continue to decrease his price, they may be out of auction after some stage.

In experimental analysis payoff using contribution and VCG schemes were obtained for randomly generated data sets. The results have been presented in following six graphs. It can be seen that:

(1) As expected, VCG discount does not change with price (Incentive Compatibility). However the payment scheme based on contribution is not strategy proof. It can be seen from graphs, that as buyer’s price decreases, his payoff also decreases (reverse for sellers). This ensures that buyer cannot make only bounded gains.

(2) The buyer’s gains are always be bounded irrespective of number of buyers and sellers changing their respective prices. The scheme always budget balanced.

(3) It can be further seen that even if more than two buyers decrease bid prices, the discount of the buyer decreases. This is due to the fact that surplus also decreases. So a buyer or set of buyers can manipulate our scheme only to limited extent, by bidding untruthfully. In other words our scheme is efficient, budget balanced and individually rational.
Figure 3.14 Effect of changes in prices on VCG Payoff and Contribution by a single buyer. (Horizontal line is VCG Payoff)

Figure 3.15 Buyer’s expected gain and changes in bid prices (Single Buyer)
Figure. 3.16 Effect of changes in VCG Payoff and Contribution Payment, when two participants change their respective prices. The payoff of one participant has been shown here.

Figure. 3.17 Change in gain when two participants change their respective prices. The payoff of one participant is shown here.
Figure 3.18 Effect of price changes by more than two buyers on VCG Payoff and Contribution Payoff of two buyers.

Figure 3.19 Change in gain of two buyers when more than two buyers change their prices.