Chapter 5

Multi Unit Auctions with Constraints

5.1 Introduction
In this chapter, we analyze the problem of the sale of a government owned enterprise under different types of constraints. We present a general formulation of this problem as an optimization problem. This formulation can handle multi unit single object auctions with different types of constraints. We show that some multi unit single object auction formulations studied previously are particular cases of our formulation. Then we develop a few results and an algorithm to obtain optimum solution based on these results. It has been shown by induction that our algorithm generates an optimum solution if it exists. It has also been shown that our algorithm can solve the problem with polynomial time complexity. The Integer Programming (IP) formulations, which are typically used to obtain optimum assignment in case of multi unit single object auctions, are hard to solve when the number of buyers or sellers is large. On the other hand, our algorithm always obtains optimum solution with polynomial time complexity. The main contribution of this work is the general formulation of the problem along with an algorithm to obtain optimum solution. Then we develop strategy proof mechanisms.

Auctions have been widely studied in economic theory. A detailed survey of auction mechanisms can be found in [W1996]. An important application of auctions in the last few years has been selling of government owned enterprises in many countries across the world. As a result of economic liberalization the government is withdrawing itself from many spheres of activity. This means that governments of different countries are reducing their shares in state owned enterprises. In such cases government has to ensure that the state owned enterprises are passed over to appropriate persons/organizations. Towards this end, it needs to design an appropriate format. In some cases, in the interest of society, the government has to ensure that a single person or an organization does not monopolize state owned assets. It has to ensure that assets are distributed appropriately. This can give rise to auctions with constraints of different types. A typical constraint can be that any individual cannot acquire ownership rights beyond a certain percentage. A group of enterprises can acquire only certain percentage of the ownership. There can be similar
restrictions on foreign investors. It may also be required that employees get a certain percentage of ownership. A common example of this type of situation is Initial Public Offerings (IPO) of a state owned enterprise. These constraints are different from the usual capacity constraints. Even though auctions have been widely studied, auctions under different types of constraints, in addition to usual capacity constraints, have not been studied in details.

Suppose that a government owns an enterprise fully or partially. When the State decides to privatize it either fully or partially, normally the process of competitive bidding is adopted. Different individuals and enterprises may bid for the purchase of equity. Even though individuals or private enterprises may indicate their capacity or willingness to buy certain number of shares, it may not be feasible as per policies of government to allocate them all the shares they bid. In the interest of the society, the government is interested in ensuring that the state owned assets are distributed appropriately and fairly and ensure that no individual, group of individuals or an enterprise monopolizes the public owned assets. These requirements give rise to different constraints, apart from usual capacity constraints. To illustrate this scenario, consider an example, where the government wishes to decrease its ownership in an enterprise from 100% to say 49% and decides to sell 1 million shares. In this case there can be several types of constraints. Suppose that the bids submitted by different buyers and enterprises are as given in Table 5.1. In this table, the third column indicates the types of buyers. It indicates that the second buyer is an employee, while the first one is a general individual buyer. The remaining five buyers belong to the enterprise category. The buyers of enterprise category have been divided into different groups. The third buyer belongs to group 1, fourth buyer belongs to group 2, fifth and sixth belong to group 3, while the last belongs to group 4. The usual capacity constraints indicating the minimum and maximum demand of individual buyers are shown in columns 4 and 5. The minimum and maximum demands of buyers 1 and 2 are 1000 and 2000 respectively.

The constraint indicating the maximum permissible limit on the individuals in the corresponding group is shown in the last column. It indicates that the upper limit for any individual buyer is 500 shares. On the other hand for any employee of the enterprise it is 1000 shares. In the same way, any enterprise in groups 1 to 3 can have maximum 1 Lakh
shares. There is an upper limit of 10000 shares each on the enterprises belonging to group 4. These constraints can limit the minimum or maximum number of shares that can be allotted to any buyer or an enterprise. In case of buyer 1, even though he has indicated his willingness to purchase 1000 to 2000 shares, the maximum shares that can be allotted, to him are 500. The scenario in case of buyers 3 to 6 is similar. These constraints restrict the number of shares that can be sold to the buyers, irrespective of the demands specified by them.

In addition to these constraints, there can be constraints on categories. For instance the total numbers of shares to be allotted to all the individual buyers should not exceed 25% of the total 1 million shares. Other restrictions can be that the total foreign equity should not be more than say 20%. These constraints are called as group constraints, since they apply to group of buyers.

<table>
<thead>
<tr>
<th>Buyer</th>
<th>Type</th>
<th>Category</th>
<th>Minimum Demand</th>
<th>Maximum Demand</th>
<th>Price</th>
<th>Upper Limit on Shares</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Individual</td>
<td>Other</td>
<td>1000</td>
<td>2000</td>
<td>105</td>
<td>500</td>
</tr>
<tr>
<td>2</td>
<td>Individual</td>
<td>Employee</td>
<td>1000</td>
<td>2000</td>
<td>100</td>
<td>1000</td>
</tr>
<tr>
<td>3</td>
<td>Enterprise</td>
<td>Group 1</td>
<td>200000</td>
<td>500000</td>
<td>105</td>
<td>100000</td>
</tr>
<tr>
<td>4</td>
<td>Enterprise</td>
<td>Group 2</td>
<td>500000</td>
<td>800000</td>
<td>107</td>
<td>100000</td>
</tr>
<tr>
<td>5</td>
<td>Enterprise</td>
<td>Group 3</td>
<td>500000</td>
<td>1000000</td>
<td>110</td>
<td>100000</td>
</tr>
<tr>
<td>6</td>
<td>Enterprise</td>
<td>Group 3</td>
<td>300000</td>
<td>1000000</td>
<td>110</td>
<td>100000</td>
</tr>
<tr>
<td>7</td>
<td>Enterprise</td>
<td>Group 4</td>
<td>10000</td>
<td>1000000</td>
<td>102</td>
<td>100000</td>
</tr>
</tbody>
</table>

An IPO with competitive bidding can be considered as an example of forward auction. In forward auction, there is a single seller having quantity q of an item to sell and there are n buyers. The objective is to maximize the selling cost or revenue. Here the government is the seller. There can be a similar problem in the reverse direction in the financial domain.

Suppose that a bank or a financial institution has some fixed amount A to invest. It invites competitive bids for placement of funds from different agencies, each of which guarantees certain return and carries certain risk depending upon the funds available.

While placing the funds, the bank has to comply with the guidelines of regulatory agencies. These guidelines may prohibit the bank from investing an amount beyond certain limit in different sectors. Suppose a bank has 50 million rupees to invest and the placement offers of different types of investments in four sectors have been received. Each has certain risk associated with it and also gives certain level of return. The bank
needs to determine the permissible amount to be invested in different sectors as well as in different types of investments. Then the objective is to minimize the risk. These can give rise to the constraints discussed earlier. Suppose different placements offers are as shown in Table – 5.2.

Table. 5.2: Offers for placement of funds

<table>
<thead>
<tr>
<th>Agency</th>
<th>Sector</th>
<th>Minimum Amount</th>
<th>Maximum Amount</th>
<th>Estimated Rate of Risk</th>
<th>Upper Limit - Sector</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sector 1</td>
<td>10 Million</td>
<td>20 Million</td>
<td>5%</td>
<td>20 Million</td>
</tr>
<tr>
<td>2</td>
<td>Sector 2</td>
<td>10 Million</td>
<td>15 Million</td>
<td>5.5%</td>
<td>10 Million</td>
</tr>
<tr>
<td>3</td>
<td>Sector 2</td>
<td>10 Million</td>
<td>25 Million</td>
<td>6.5%</td>
<td>10 Million</td>
</tr>
<tr>
<td>4</td>
<td>Sector 3</td>
<td>15 Million</td>
<td>25 Million</td>
<td>7%</td>
<td>15 Million</td>
</tr>
<tr>
<td>5</td>
<td>Sector 3</td>
<td>15 Million</td>
<td>25 Million</td>
<td>6%</td>
<td>15 Million</td>
</tr>
<tr>
<td>6</td>
<td>Sector 4</td>
<td>1 Million</td>
<td>10 Million</td>
<td>8%</td>
<td>5 Million</td>
</tr>
<tr>
<td>7</td>
<td>Sector 1</td>
<td>20 Million</td>
<td>30 Million</td>
<td>6%</td>
<td>20 Million</td>
</tr>
</tbody>
</table>

It can be seen from the above table that the last column indicates the sector wise limits to be complied by the bank and hence different constraints arise. On the other hand, the problem of placement of funds is similar to the reverse auction problem, in the sense that bank is the buyer and placement agencies are sellers. In reverse auction, there is one buyer who requires quantity q of a certain item and there are n sellers, who can supply these items. The objective is to procure the items at the minimum cost. Reverse auctions are helpful in procurement. These types of constraints can arise there too. Consider a scenario where the government may ask the enterprises to procure certain quantity of items from the cooperative organizations run by groups of persons from economically weaker sections which manufacture the same items.

The present day electronic auctions support novel applications like electronic procurement, bidding on air ticket etc. Different companies use electronic bidding to set prices of their goods. A multi attribute auction system, for electronic procurement, has been studied in [BKS1999]. In multi attribute auctions, the winner determination is based on more than one attribute unlike in traditional English or Dutch auctions, where price is the only attribute. An application of auction theory in electronic procurement has also been studied in [EGKL2001], and it also gives a near optimal solution to bid evaluation problem of the buyer. A procurement process, which minimizes the cost of
procurement using auctions has been proposed in [KL2001]. Another type of auction known as Combinatorial Auctions where seller wishes to sell a combination of goods and buyers bid on one or more goods, has been studied recently [RPH1998] [S2002]. A survey of combinatorial auctions problem can be found in [VV2003]. An application of auction theory in transportation services is presented in [LOPST2002]. An approximately-strategy proof and tractable multi unit auction mechanism for single good multi unit allocation problem has been presented in [KPS2003]. It presents a fully polynomial time approximation scheme for reverse and forward auction variations. We formulate this problem as a mixed integer programming problem and develop an algorithm based on branch and bound method. Later we also obtain the VCG Payoff [V1961], [C1971], [G1973]. It can also handle formulation presented in [KPS2003]. The constraints considered in this work are different from the budget constraints or liquidity constraints studied earlier [BCIMS2005]. In budget constraints buyers have constraint on available budget. These constraints restrict the total quantity that a buyer can purchase. The constraints considered in the present work are basically assignment constraints, which restrict quantity to be allocated to a buyer or group of buyers. In the present work it is assumed that buyers do not have liquidity constraints. In our formulation buyer can specify the maximum or the minimum quantity required. Our formulation can handle liquidity constraints by means of minimum and maximum quantities. We formulate IPO auction problem as single object multi unit problem with constraints. Further we develop an algorithm based on branch and bound method. Later we also obtain the VCG Payoff [V1961], [C1971], [G1973]. It can also handle formulation presented in [BVSV2002]. Our formulation allows buyers to submit bids in different formats. Apart from this, we are not aware of similar work in financial domain requiring different types of constraints.

5.2 Problem Formulation and Results
In this section the problem of optimal matching in case of IPO is formulated as multi unit auction problem with different types of constraints. This problem is formulated as non linear integer programming problem. Then a few specific cases are discussed. Our formulation can handle forward as well as reverse auctions. In IPO the State is only seller and there are a number of different buyers. This can be considered as multi unit forward auctions with different types of constraints. In forward auctions, the objective is to
maximize the cost of selling whereas in reverse auction the objective is to minimize the cost of procurement. Even though it has been formulated in the context of IPO, the formulation is general and can handle any type of multi unit auctions with constraints. Our reverse auction problem for minimization of risk is symmetric and can address any similar problem. The multi unit auctions with constraints have been considered in different settings here.

The concepts of bids and asks are already introduced in the 3rd chapter, in the context of double auctions. The same concept is used here. A bid or an ask is also an ordered list of attribute names and values. A bid in the multi unit auction (a bid in forward auction and ask in reverse auction) describes the details of the items (shares in the context of IPO), its quantity and price that the buyer is willing to pay. Additionally a buyer (seller) can describe capacity constraints. Each $B_i(A_i)$ is of the following type.

$$B_i = (v_{i1}, v_{i2}, \ldots, v_{ik})$$

where $v_{ij}$ is the value of the $j^{th}$ attribute. The price and quantity are two attributes of asks and bids. The attributes describe different characteristics of the items. Each attribute assumes values from the set of specified domains. For instance, the price attribute will have values from set of positive real numbers. In this work price and quantity attributes are considered. Let there be $n$ bids, $B_1, B_2, \ldots, B_n$. Let $BD$ be the set of all bids. In IPO buyers can submit bids in a number of different ways. It is always assumed that the prices in bid refer to per unit price.

(1) A buyer can submit a single price-quantity pair, indicating the amount he is willing to pay for the corresponding quantity. Additionally a buyer can indicate whether it is “all or nothing bid”. “All or nothing” bid means that the buyer requires the entire quantity specified and no reduction in the quantity to be purchased is acceptable. If “all or nothing” is not specified then the meaning of the bid is that the buyer has specified the maximum quantity he is willing to buy at the specified price. The price, quantity and “all or nothing” indication in the bid are of the form $\{(bp_i, bq_i), \text{ all-or-nothing }, \ldots \}$ or $\{(bp_i, bq_i), \text{ max-qty }, \ldots \}$. We call these bids as bids of type 1.

(2) Alternatively a buyer can submit a set of quantity ranges and corresponding prices. A quantity range is indicated by a quantity interval. The quantity interval
is closed at the higher end. This bid can be interpreted as buyer is willing to pay the stated price for any quantity in that interval. The price and quantity intervals (but assumes integer values) in the bid is of the form

\{((bq_1, bq_2], bp_1), ((bq_3, bq_4], bp_2), \ldots \}.

So buyer is willing to pay \(bp_1\) per unit for any quantity between \(bq_1\) (excluding) and \(bq_2\) (including) and \(bp_2\) per unit for any quantity between \(bq_3\) (excluding) and \(bq_4\) (including). If \(bq_1 > 0\), it means that buyer requires minimum quantity of \((bq_1+1)\). In this case \(bp_i > bp_{i+1}\) and \(bq_i < bq_{i+1}\) for all \(i\). Only one value for quantity is selected in the optimum assignment. We call these bids as bids of type 2.

(3) In another option a bid can also be submitted as a set of price, quantity pairs indicating the marginal price buyer is willing to pay for additional units. The price, quantity and all or nothing indication in the bid are of the form

\{(bp_1, bq_1), all-or-nothing, (bp_2, bq_2), (bp_3, bq_3), max-qty, \ldots \}.

The interpretation of this bid is that buyer is willing to pay \(bp_1\) per unit for quantity \(bq_1\). It is all or nothing type of bids. So it is the minimum quantity required. Further, buyer is willing to pay \((bp_1+bp_2)\) per unit for \(bq_2\) more units. However, these units can be acquired only after getting first \(bq_1\) units. In the subsequent price quantity pair buyer can specify only maximum quantity. After obtaining \(bq_1\) units, buyer can get \(bq_2\) more units. The maximum quantity that can be acquired by the buyer is \((bq_1+bq_2+bq_3)\). It further states that buyer is willing to pay \(bp_1+bp_2+bp_3\) per unit for additional unit after \((bq_1+bq_2)\). We call these bids as bids of type 3.

In our set up these different options are supported. These bids and asks are processed electronically. We have formulated IPO auction as multi unit single object auction with constraints. This formulation can handle both forward and reverse auctions. We also develop some new results that help in devising an efficient algorithm.

As already stated three different options are supported in our formulation. These can handle many practical situations. These cases are described earlier. In second case (bid of type 2) buyer can also use marginal piecewise bidding language for submitting bids [KPS2003]. If a buyer (seller) uses marginal decreasing piecewise bidding language, then
\( v^i \) can be a semi closed interval, which is open at one of the ends. Each buyer (seller) can specify \( m \) such intervals and price pairs. In this case we select only one point (on an interval) for each buyer. In the third case, buyer can submit bids as set price quantity pairs, where each pair indicates marginal price buyer is willing to pay for additional units. In case there are no assignment constraints and all buyers specify unit quantity, then it is usual single object multi unit auction. First we formulate the optimization problem and then discuss few particular cases.

Let us define
\[
x_{ij} = \begin{cases} 
1 & \text{if } j^\text{th} \text{ quantity interval is selected for } i^\text{th} \text{ bid} \\
0 & \text{otherwise}
\end{cases}, \quad i = 1, 2, \ldots, n \text{ and } j = 1, 2, \ldots, m
\]

If \( k^\text{th} \) buyer (seller) specifies only price, quantity pair, then \( m = 1 \) for \( i = k \).

Let \( q_{ij} \) be the quantity purchased by the \( i^\text{th} \) buyer at price \( b_{p_{ij}} \). Then we define our optimization problem as follows
\[
\max \sum\sum x_{ij}q_{ij}b_{p_{ij}} \quad (5.1)
\]

The different constraints are defined as follows.

Quantity Constraint: It is the upper limit on the total quantity to be purchased.
\[
\text{Total quantity} = \sum\sum q_{ij} \leq q, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, m \quad (5.2)
\]

Capacity Constraint: The capacity constraint indicates the minimum and maximum demand of the buyer. In some cases, these constraints may be implicitly specified. Let \( b_{q_{\text{max}}} \) be the maximum quantity required by \( i^{\text{th}} \) buyer. Then capacity constraint of buyer \( i^{\text{th}} \) buyer is
\[
\sum q_{ij} \leq b_{q_{\text{max}}}, \quad i = 1, 2, \ldots, n \quad (5.3)
\]

If \( i^{\text{th}} \) buyer specifies \( b_{q_{\text{min}}} \), as minimum required quantity, then we have constraint
\[
b_{q_{\text{min}}} \leq \sum q_{ij} \quad (5.4)
\]

In case the minimum required quantity is specified, then that minimum quantity must be allocated to that buyer, in case his bid is selected. In case \( i^{\text{th}} \) buyer specifies all or nothing bid, the capacity constraint for \( i^{\text{th}} \) buyer changes as
Group Constraint: Let \( A \) be the set of buyers, such that there is a limitation on total quantity that can be purchased by a group of buyers. There can be more than one such set. Let \( q_a \) be the quantity, then we have

\[
\sum_{i} \sum_{j} q_{ij} \leq q_a, \text{ where } i \in A \text{ for all such groups}\]  (5.5).

The other constraints are

\[
\sum_{j=1}^{m} x_{ij} \leq 1 \text{ for all } i \in B \text{ and all } q_{ij} \text{ are non negative integers. In case any bid is of type 3, the constraint changes to} \]

\[
x_{ij} = 0 \text{ or } 1 \ , \ x_{ij} \leq x_{ij+1} \text{ and all } q_{ij} \text{ are non negative integers (for all } i \text{ and } j). \]

\[
bql_{ij} < q_{ij} \leq bqu_{ij} \ , \text{ where } bqu_{ij} \text{ and } bql_{ij} \text{ are the upper and lower limits of } j^{th} \text{ price quantity interval submitted by } i^{th} \text{ buyer.} \]  (5.6)

The first constraint in (5.6) ensures that only a single point in any interval is selected for buyers with bids of types 1 and 2. The second constraint ensures that quantity is an integer. The last constraint ensures that quantity allocated is within the corresponding interval. The optimization problem defined in (5.1) is basically nonlinear integer-programming problem. This happens because quantity is also decision variable. However it assumes only integer values. So both decision variables \((x_{ij} \text{ and } q_{ij})\) assume integer values. This formulation can handle the case where a few buyers have submitted only price quantity pairs and others have submitted intervals. We state a few particular cases of our general formulation. In these cases our basic algorithm can be modified.

### 5.3 Particular Cases

In case of reverse auction the formulation changes as

\[
\min \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} q_{ij} p_{ij} \]

The different constraints are defined as follows

Quantity Constraint: It is the upper limit on the total quantity to be sold

\[
\text{Total quantity } = \sum_{i=1}^{n} \sum_{j=1}^{m} q_{ij} \leq q \ , \ i = 1, 2, ..., n \ , \ j = 1, 2, ..., m \]

Capacity Constraint: The capacity constraint indicates the minimum and maximum supply by the seller. In some cases these constraints may be implicitly specified. Let
aq_{imax} be the maximum quantity available with i^{th} seller. Then capacity constraint of i^{th} seller is \( \sum_{j=1}^{n} q_{ij} \leq \ aq_{imax} \quad i = 1, 2, ..., n \)

If i^{th} seller specifies aq_{imin}, as minimum quantity to be sold, then we have constraint\[ aq_{imin} \leq \sum_{j=1}^{n} q_{ij} \]

In case minimum quantity to be sold, is specified by a seller, then that minimum quantity must be purchased from that seller, in case his ask is selected.

Group Constraint: Let A be the set of sellers, such that there is limitation on total quantity that can be purchased from them. There can be more than one such set. Let qa be the quantity, then we have \[ \sum_{i \in A} \sum_{j=1}^{n} q_{ij} \leq qa, \text{where } i \in A \text{ for all such groups} \]

The other constraints are \[ \sum_{j=1}^{n} x_{ij} \leq 1 \quad \text{for all } i \in B \text{ and all } q_{ij} \text{ are non negative integers.} \]

In case any bid is of type 3, the constraint changes to \[ x_{ij} = 0 \text{ or } 1, \quad x_{ij} \leq x_{ij+1}, \text{all } q_{ij} \text{ are non negative integers (for all } i \text{ and } j). \]

aq_{ij} < q_{ij} \leq aqu_{ij}, \text{ where } aqu_{ij} \text{ and } aq_{ij} \text{ are the upper and lower limits of } j^{th} \text{ price quantity interval submitted by } i^{th} \text{ seller.}

**All bids of type 1**

In case all buyers submit set of price quantity pairs then m = 1 for all the buyers. In this case second subscript j can be omitted. So the above formulation gets changed as follows

\[ x_{i} = \begin{cases} 1 & \text{if } i^{th} \text{ bid is selected} \\ 0 & \text{otherwise} \end{cases}, \quad i = 1, 2, ..., n \]

if i^{th} bid is selected

Let q_{i} be the quantity sold to the i^{th} buyer at price b_{pi}. Then we define our optimization problem as follows

\[ \max \sum_{i=1}^{n} x_{i}q_{i}b_{pi} \quad (5.7) \]

The constraints are defined as follows

The quantity, capacity and group constraints are similarly defined. These constraints are
\[ \sum_{i=1}^{n} q_i \leq q, \quad i = 1, 2, \ldots, n, \quad m = 1 \]  \hspace{1cm} (5.8)

\[ q_i \leq b_{q_{\text{max}}} \quad i = 1, 2, \ldots, n \]  \hspace{1cm} (5.9)

\[ b_{q_{\text{min}}} \leq q_i \]  \hspace{1cm} (5.10)

If \( i^{\text{th}} \) buyer specifies all or nothing bid then above constraint changes as

\[ q_i = b q_i \]  \hspace{1cm} (5.11)

\[ \sum q_i \leq q_a, \text{ where } i \in A \text{ for all such groups} \]  \hspace{1cm} (5.12)

The other constraints are \( q_i \) are nonnegative integers for all \( i \) and \( x_i = 0 \) or 1 for all \( i \).

The optimization problem defined above is a particular case of more general formulation defined in (5.1). However it is still a nonlinear integer programming problem. In case all bids are all or nothing bids, then the formulation reduces to 0-1 programming problem. In this case the variable \( x \) can be omitted and the problem becomes

\[ \max \sum_{i} q_i b p_i \]  \hspace{1cm} (5.13)

The constraints are as follows.

Quantity Constraint: Total quantity \( \sum q_i \leq q, \quad i = 1, 2, \ldots, n \)  \hspace{1cm} (5.14)

Capacity Constraint: \( q_i = b q_i \) for all \( i = 1, 2, \ldots, n \)  \hspace{1cm} (5.15)

These constraints capture all or nothing bids.

Group Constraint: Let \( A \) be the set of buyers, such that there is a limitation on the total quantity that can be purchased (sold) to them. There can be more than one such set. Let \( q_a \) be the quantity, then we have

\[ \sum q_i \leq q_a, \text{ where } i \in A \text{ for all such groups} \]  \hspace{1cm} (5.16)

The other constraint is \( q_i \) are non negative integers for all \( i \)  \hspace{1cm} (5.17)

The formulation for reverse auction problem can be similarly defined and it is omitted.

Apart from these two cases, the case - where all bids are marginal prices (i.e. type 3) are submitted - is considered later. The formulation presented in [BVSV2002] and usual single object multi unit auctions are particular cases of this formulation. It can be seen that our formulation is general and can handle different cases. Our algorithm is general one and can handle these different cases.
As stated earlier, the two problems addressed in this work are as follows.

(i) Determining optimum assignment of bids and asks.

(ii) Determining how much each participant has to pay or receive.

We first obtain its optimum solution and then obtain VCG Payoff. VCG Payoff is obtained first by obtaining optimum solution of (5.1) with all buyers (sellers) and then solving the optimization problem by removing a buyer (seller). Let \( V_o \) be the optimum solution to (5.1). Let \( V_{oi} \) denote the optimum solution after removing \( i^{th} \) buyer (seller).

Then VCG Payoff of \( i^{th} \) buyer is

\[
pb_i = bp_i - (v_o - v_{oi}), \text{ similarly for } i^{th} \text{ seller } sp_i = as_i + (v_o - v_{oi})
\]

where as \( bp_i, as_i \) are bid and ask prices specified by \( i^{th} \) buyer or seller respectively. The LHS represents the amount payable. The VCG Payoff ensures that truthful bidding is the dominant strategy and hence buyers bid truthfully. We compute VCG Payoff without actually solving a series of optimization problems. Later on we work to generalize uniform price auction.

Assignment: An assignment of quantity \( qa_{ij} \) at price \( bp_{ij} \) from \( i^{th} \) bid \( B_i \) is denoted by \( (bp_{ij}, qa_{ij}) \). In this case \( bp_{ij} \) is the price component of \( j^{th} \) price quantity pair of bid \( B_i \). The assigned quantity \( qa_{ij} \) satisfies the relation, \( \max(0,bq_{ij-1}) \leq qa_{ij} \leq bq_{ij} \) and \( bq_{i0} = 0 \). The terms \( bq_{ij} \) and \( bq_{ij-1} \) are quantity components of \( j-1 \) and \( j^{th} \) price quantity pairs of bid \( B_i \).

Contribution: We define contribution to the value of objective function at price \( bp_i \) and quantity \( bq \) as \( \text{cov}_{pi}(bq) \) as follows

\[
\text{cov}_{pi}(q) = bp_ibq
\] (5.18)

The contribution results from assignment of quantity \( bq \) at price \( bp_i \). The value of (5.18) is the maximum (minimum) for any \( bp_i \), when \( q \) is the maximum (minimum). In the same way for any \( bq \) the value is maximum (minimum) when \( bp_i \) is maximum (minimum).

Using these we obtain the upper bounds on improvement at any stage. If there is only one bid (ask) at price \( bp_i \), then we call it as buyer’s (seller’s) contribution and indicate it by \( \text{cov}_{bi}(bq) \). If \( bq = 1 \), we call it as unit contribution and indicate it by \( \text{ucov}_{pi} \).

Combination: Let there be \( n \) bids \( B, B_2, B_3, \ldots, B_n \). We define combination for quantity \( q \), as set of price quantity pairs \( (bp_{ij}, qa_{ij}) \) such that each pair represents assignment of quantity \( qa_{ij} \) at price \( bp_{ij} \) from bid \( B_i \). It is a set \( \{(bp_{ij}, qa_{ij}), j = 1, 2, \ldots, m, i = 1, 2, \ldots, \)
k}. In case $i^{th}$ bid is of type 1 or 2, then $m = 1$ for such bids. The quantities satisfy the relation
\[ \sum_{i=1}^{k} \sum_{j=1}^{m} qa_{ij} = q. \]

In other words a combination represents a set of assignments from bids $B_1, B_2, B_3, \ldots, B_k$. A combination is called a feasible combination, if it satisfies all the constraints of assignment. In case $i^{th}$ bid $B_i$ is submitted with marginal piecewise bidding language, then a feasible combination has only a single pair from bid $B_i$. In other words in this case $m = 1$. The value of the combination $C_i$ for quantity $q$, $\text{val}_i(q)$ is defined as
\[ \text{val}_i(q) = \sum_{i=1}^{k} \sum_{j=1}^{m} qa_{ij}bp_{ij}. \]

A feasible combination $C_j$ is optimum combination for quantity $q$ if $\text{val}_j(q) \geq \text{val}_i(q)$, (for all other feasible combinations). The value of the optimum combination is the maximum. In order to obtain optimum assignment for (5.1), an optimum combination is required to be created. We now show how such combination can be created. We begin with a combination with single assignment.

**5.4 Creating an Optimum Combination**

Upper bound on improvement at each stage: Let maximum price among the received bids be $bp_1$. Let $q_1$ be the maximum quantity that can be assigned at this price. Then it can be seen that the maximum possible improvement or upper bound on improvement at any stage is
\[ \text{cov}_1 = bp_1 q_1 \quad (5.19). \]

Optimum assignment for a fixed quantity: A buyer’s bid can have a single price quantity pair $(bp_i,bq_i)$. Alternatively a bid can be submitted by using the marginal decreasing piecewise bidding language. In this case, the buyer specifies a list of quantity ranges and prices he is willing to pay $(bp_{i1},(bq_{i0},bq_{i1}))$. This can be interpreted as the buyer is willing to pay price $bp_{i1}$, for quantity $q$, where $q \in (bq_{i0},bq_{i1}]$ and assumes integer values. We call the list of prices and quantity intervals as price, quantity part of the bid. The price, quantity part of a bid (ask) denoted by $B_{pi} (A_{pi})$ is ordered list of price and quantity intervals as

196
\[ B_{pi} = \{(bp_{i1}, (bq_{i0}, bq_{i1})), (bp_{i2}, (bq_{i1}, bq_{i2})), (bp_{i3}, (bq_{i2}, bq_{i3})), \ldots, (bp_{im}, (bq_{im-1}, q_{in})}\}. \]

The prices and quantities satisfy the following relations

\[ bp_{ij} > bp_{ij+1} \text{ for all } i \text{ and } j \quad \text{and} \quad q_{ij} < q_{ij+2} \text{ for all } i \text{ and } j. \]

We call \( bp_{ik} \) as price component of \( k^{th} \) element of \( B_{pi} \) and \( (bq_{ik-1}, bq_{ik}] \) as the quantity interval of \( k^{th} \) element. In further discussion in this section, without loss of generality, we state further discussion for the bids. However it can be easily extended for asks. So we do not explicitly state it for ask.

Only a pair of price and quantity \( (bp_{ik}, bq_{ik1}) \), called as the point, from any one element (say \( k^{th} \) element) of the set \( B_{pi} \) can be selected for each buyer in case of an optimum assignment. In this case \( p_{ki} \) is the price of the \( k^{th} \) element and \( q_{k1} \) is a point belonging to the corresponding quantity interval and assumes integer values.

In order to obtain optimum assignment for quantity \( q \), we consider two cases. In the first case a buyer submits only price and quantity pair and no intervals are submitted. In this case it is assumed that all buyers submit only price quantity pair. In second case one or more quantity intervals are submitted along with corresponding price. It is also possible here that a few buyers may submit only price quantity pairs.

In order to obtain optimum solution we carry out the assignment in stages, till the assignment of quantity \( q \) is completed. In first case it is always possible to determine the maximum price and the quantity available at that price, at any stage. In the second case it is not always possible to get such an assignment. So we create different price quantity combinations and then determine the optimum assignment.

**Case 1:** Let \( B_1 \) be the bid with the maximum unit contribution and price \( bp_1 \). Let it be \( ucov_1 \). Let \( q \) be the maximum quantity that can be assigned at this price. Let \( cov_1(q) \) be the contribution from this assignment. Then

\[ cov_1(q) = (q) \times (ucov_1) \]  

(5.20)

This is the maximum possible improvement from an assignment of quantity \( q \). 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, term \( ucov_{i1} \) (i.e. unit contribution) is improved or \( q \) (assigned quantity) is improved. In this case neither is possible, as there is no bid with higher unit contribution and quantity assigned is the maximum available. So expression in (5.20)
represents, the maximum possible improvement from single assignment. In case, there are more than one bid with maximum unit contribution, then we combine respective bid quantities. Due to this, we find out maximum quantity that can be assigned for that unit contribution.

Suppose that the bid $B_1$ with maximum unit contribution has only quantity $q_1 (< q)$ available. Suppose that $B_2$ is another bid with bid price $p_2$ and having the highest contribution to value of objective function by assignment of unit quantity except ask $B_1$. Let $u_{cov_2}$ be the contribution to the value of objective function by assignment of unit quantity of bid $B_2$. Then we have

$$u_{cov_2} > u_{cov_j} \text{ for } j = 3, \ldots, n \text{ and } u_{cov_2} < u_{cov_1}$$

(5.21)

Let $q_2$ be the maximum quantity that can be assigned at price $p_2$ such that $q_1 + q_2 = q$. Then these two assignments represent the maximum improvement that can be obtained by assignment of quantity $q$. This maximum contribution to the value of objective function is

$$cov_1 + cov_2 = (q_1) \times (u_{cov_1}) + (q_2) \times (u_{cov_2})$$

(5.22)

It can be seen that contribution can be improved if any one of the two terms on right hand side can be improved. As already seen the first term cannot be improved. The same argument can be applied to the second term, as $B_1$, the only bid which can improve unit contribution of bid $B_2$, is already assigned. There is no other bid with higher unit contribution. The quantity assigned is also the maximum possible. As the second term cannot be improved so (5.22) is the highest possible improvement in this stage. In this case also we combine bids with same unit contribution.

**Theorem 1**

Let $BD_{k-1} = \{B_1, B_2, \ldots, B_{k-1}\}$ be the set of bids which can be assigned and satisfy the following condition.

1. For any bid $B_j \in BD_{k-1}$, $u_{cov_j} > u_{cov_{j+1}}, j = 1, 2, \ldots, k-2$. Let the $q_1, q_2, \ldots, q_{k-1}$ be the quantities assigned respectively.

2. For any bid $B_i \not\in BD_{k-1}$, $u_{cov_i} < u_{cov_1}$, where $l = 1, 2, \ldots, k-1$ (i.e. set of bids belonging to $BD_{k-1}$).

3. Let $B_k$ be the bid such that $u_{cov_k} > u_{cov_j}$, where $j = k+1, \ldots, n$ (set of bids not in $BD_{k-1}$). The assignment of quantity $q_k$ of this bid completely satisfies the demand of quantity $q$.  

198
Then maximum improvement can be obtained by a combination consisting of all bids in BD_{k-1} and B_k. This is the maximum possible improvement by assigning set of bids for quantity q.

**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 to the value of objective function tcov_k after assignments of k bids is

\[
tcov_k = \sum_{i=1}^{k-1} (q_i) x (ucov_{i}) + (q_k) x (ucov_k)
\]  

(5.23)

The contribution can be improved, if and only if the second term can be improved. This is due to the fact that the first term cannot be improved by assumption. In this case, it is not possible to improve the second term since its unit contribution ucov_k is the highest among the remaining bids (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. Hence tcov_k is the maximum improvement in the value of objective function. So the result follows by induction.

**Theorem 2:** Suppose that there is a bid B with price p and the maximum quantity q and the following conditions are satisfied

1. There is no bid with price higher than p (i.e. it is the highest price bid).
2. No other single bid exists with quantity q.
3. There are k bids B_1, B_2, ..., B_k, with prices p_i and quantities q_i such that \( q = \sum q_i \) and \( p_i < p \) for all i. All the remaining bids have lower price than p_k.

Then bid B has higher contribution than total contributions of bids B_1, B_2, ..., B_k.

**Proof:** Let cov_b be the contribution of bid B. Then

\[
cov_b = pq
\]

Let cov_{bi} be the contribution of bid B_i and tcov_{Bk}, be the contribution of bids B_1, B_2, ..., B_k.

\[
cov_{bi} = p_i q_i \quad \text{and} \quad tcov_{Bk} = \sum p_i q_i
\]

\[
cov_{bi} - tcov_{Bk} = pq - \sum p_i q_i = p \sum q_i - \sum p_i q_i
\]
\[ \sum_{i=1}^{k} p_i q_i = \sum_{i=1}^{k} (p_i - p_i) q_i > 0, \text{ as } q_i > 0 \text{ and } (p - p_i) > 0 \text{ for all } i. \]

**Case 2:** In case a buyer has submitted a set of price-quantity intervals (using marginal decreasing piecewise bidding language) it may not be possible to get a set of bids as in case 1. Here, at any stage of assignment only one point (called as price, quantity combination) can be selected for a buyer. We start with a scenario where two bids (each having set of price quantity intervals) are required to be combined to fulfill supply of quantity q. Since the case with one bid is similar to the earlier we do not consider it here.

Suppose that there are two bids \( B_1 \) and \( B_2 \) such that

\[
B_{p1} = \{(p_1,(q_0,q_1)), (p_3,(q_1,q_3))\} \text{ and } \]
\[
B_{p2} = \{(p_2,(q_0,q_2)), (p_4,(q_2,q_4))\}
\]

(1) The following relations are satisfied \( p_1 > p_3, \ p_1 > p_2, \ p_2 > p_4, \ p_3 > p_2 \)

(2) The quantities satisfy \( q_1 + q_2 = q \) and \( q_1 < q_3, \ q_3 < q \) and \( q_2 < q_4 \).

(3) All other bids have prices lower than \( p_4 \).

The assumptions \( p_1 > p_3, \ p_2 > p_4, \ q_1 < q_3 \) and \( q_2 < q_4 \) follow from the definition of marginal decreasing cost function and if \( p_3 > p_2 \) is not assumed this case reduces to earlier scenario and \( p_1q_1+p_2q_2 \) the optimum assignment for quantity q (from earlier results). We discuss the case where \( q_3 > q \) later (combination 5 in the Table 5.3).

<table>
<thead>
<tr>
<th>Serial Number</th>
<th>Combination</th>
<th>Contribution to the Value of Objective Function</th>
<th>Quantity (Total =q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( p_1,q_1 ) and ( p_2,q_2 )</td>
<td>( p_1q_1+p_2q_2 = (\text{c01}) )</td>
<td>( q = q_1+q_2 )</td>
</tr>
<tr>
<td>2</td>
<td>( p_1,(q_1-1) ) and ( p_4,(q_2+1) )</td>
<td>( p_1q_1+p_4q_2-p_1+p_4 = (\text{c02}) )</td>
<td>( q = (q_1-1)+(q_2+1) )</td>
</tr>
<tr>
<td>3</td>
<td>( p_3,q_3 ) and ( p_2, (q-q_3) )</td>
<td>( p_3q_3 + p_2q - p_2q_3 = (\text{c03}) )</td>
<td>( q = q + q_3 - q_3 )</td>
</tr>
<tr>
<td>4</td>
<td>( p_3,q_3 ) and ( p_4, (q-q_3) )</td>
<td>( p_3q_3 + p_4q - p_4q_3 = (\text{c04}) )</td>
<td>( q = q + q_3 - q_3 )</td>
</tr>
<tr>
<td>5</td>
<td>( p_3,q )</td>
<td>( p_3q = (\text{c05}) )</td>
<td>If ( q_3 &gt; q )</td>
</tr>
</tbody>
</table>

It can be easily seen that \( q_1 + q_4 \geq q \) (as \( q_1 > q_3 \)). The possible combinations for quantity q with contribution to the values of objective functions are It can be seen that

\[ \text{co1} - \text{co2} = p_1q_1+p_2q_2 - p_1q_1 - p_4q_2 + p_1 - p_4 = (p_2-p_4)q_2 + p_1 - p_4 > 0 \text{ as } p_2 > p_4 \text{ and } p_1 > p_4. \]

\[ \text{co3} - \text{co4} = p_3q_3 + p_2q - p_2q_3 - p_3q_3 + p_4q + p_4q_3 = (p_2-p_4)q - (p_2-p_4)q_3 = (p_2-p_4)(q-q_3) > 0 \text{ as } p_2 > p_4 \text{ and } q > q_3. \]
So combinations (2) and (4) (in Table – 5.3) cannot be optimum assignment for quantity q. It can be further seen that, by decreasing quantity assigned at higher price \( (p_1 \) or \( p_3 \)) or by increasing quantity assigned at lower price, the contribution to the value of objective function cannot be improved. Out of above five combinations, combinations 1,3 or 5 can be possibly optimum.

Then the conditions under which combination (3) is optimum assignment are

\[
\begin{align*}
 p_3q_3 + p_2q_2 - p_2q_3 &> p_1q_1 + p_2q_2 \\
 p_3q_3 + p_2(q_1 + q_2) - p_2q_3 &> p_1q_1 + p_2q_2 \\
 p_3q_3 + p_2(q_1 - q_3) &> p_1q_1
\end{align*}
\]

(5.24)

It can be seen that if (5.24) is satisfied, then combination (3) (in Table –5.3) is optimum for quantity q, otherwise one of the combinations, (1) or (5) is optimum. It can be seen that combination (5) is optimum if

\[
\begin{align*}
 p_3q &> p_1q_1 + p_2q_2 \\
 p_3(q_1 - q_2) &> p_1q_1 + p_2q_2 \\
 (p_3 - p_2)q_2 &> (p_1 - p_3)q_1
\end{align*}
\]

(5.25)

If condition (5.25) is satisfied then combination (5) is optimum, otherwise combination (1) is optimum. Using this we obtain optimum assignment in different cases.

Suppose that there are n buyers. Each buyer has submitted a bid consisting of m quantity intervals. The price, quantity part of the bid (ask) of \( i^{th} \) buyer denoted by \( B_{pi} \) is an ordered list of price and quantity intervals as

\[
B_{pi} = \{(p_{i1},(q_{i0},q_{i1}]], (p_{i2},(q_{i1},q_{i2}]], (p_{i3},(q_{i2},q_{i3}]], \ldots, (p_{im},(q_{in-1},q_{in}])\}.
\]

A combination \( C_i \) is a set of price quantity pairs \( (p_{ij},q_{ji}) \), such that \( p_{ij} \) is the price component of \( j^{th} \) element of \( B_{pi} \) and \( q_{ji} \) is the point from corresponding quantity interval. A combination has property that it can have at most one point \( (p_{ij},q_{ji}) \) from \( B_{pi} \) in case if bids of type 1 and 2. The contribution of combination \( C_i \) is the value of the combination, which is already defined. These two terms mean the same thing in the context of a combination. An assignment \( (p_{ij+k},q_{ij+k}) \), can be added to the combination \( C_i \) with contribution \( c_{ij} \) and quantity \( q_i \) if \( k \neq 0 \) and there is a bid with price quantity interval \( (p_{ij+k},[q_{ij+k-1},q_{ij+k}]) \). Then the contribution of combination becomes \( c_{ij} + p_{ij+k}q_{ij+k} \) and quantity becomes \( q_i + q_{ij+k} \). We state theorems (3) and (4) in case where there is no single highest price bid for quantity q.
**Theorem 3:** Let $C_i$ be the combination with contribution to the value of objective function $c_{oi}$, which is optimum for quantity $q_i$. Suppose that an assignment $(p_i, q)$ can be added to this combination. Further suppose that this assignment has the highest possible contribution for quantity $q$. Then combination $C_j$, consisting of $C_i$ and an assignment $(p_j, q)$ is optimum combination for quantity $q_i+q$ with optimum contribution to the value of objective function $c_{oi}+p_jq$.

**Proof:** The result can be shown to be true for $k = 1$. It can be easily verified that a combination with a single assignment $(p_i, q)$ has optimum contribution $p_iq$ for quantity $q$, when $p_i$ is maximum. Suppose that there is a combination $C_i$ consisting of $k$ assignments, which is optimum for quantity $q_i$ with contribution $c_{oi}$. Suppose that the assignment $(p_i, q)$ can be added to this combination. The price $p_i$ is the highest for quantity $q$. After adding this point to combination $C_i$ the contribution of new combination is

$$c_{oi} + p_iq$$

and quantity is $q_i+q$.

It can be easily seen that (5.26) is the optimum value for quantity $q_i+q$, as neither the first term nor the second term of (5.26) can be improved. So the result follows by induction.

**Theorem 4:** Suppose that there exist a set of $k$ combinations $C_1, C_2, \ldots, C_k$ consisting of different price-quantity assignments from bids submitted by different buyers. Let $c_{oi}$ be the contribution of each combination and $q_i$ be the respective quantities. Each combination is optimum for corresponding quantity $q_i$. The following conditions are satisfied by these combinations

1. Let $q$ be the quantity such that $\sum q_i < q$ and also $q_i < q$ for all $i$.
2. Apart from $C_1, C_2, \ldots, C_k$ no other optimum combination can be formed for any quantity $q_i < q$ from the submitted bids.
3. Suppose that $(p_j, q_j)$ is an assignment with the highest price $p_j$, which can be added to each of above combination. After adding this assignment, quantity of each combination becomes $q_i+q_j > q$ for all $i$. Let us define $q_{rem} = q - q_i$. Let $q_{remm} = \max(q_{rem})$. Suppose that there is no price-quantity assignment having price higher than that of $p_j$ and quantity smaller than $q_{remm}$.
Then the combination formed by adding point \((p_j, qtrem_i)\) to \(C_i\) having maximum contribution is an optimum combination for quantity \(q\).

**Proof:** After adding point \((p_j, qtrem_i)\) to each combination, the contributions of each combination is

\[c_0 + p_jqtrem\]

The quantity of each combination is \(q\). The combination with \(\max(c_0 + p_jqtrem)\) has the highest contribution. So it is optimum combination for quantity \(q\) (among \(C_1, C_2, \ldots, C_k\)). In case, there is more than one such combination, any combination can be selected.

We further show that apart from these combinations, there is no combination, which can be formed from the given bids and having optimum contribution for quantity \(q\).

It can be seen from result (1) and (2), that we can obtain optimum value for any quantity by selecting (1) the highest price bid with quantity \(q\), (2) selecting combination of optimum bids. In this case, scenario (1) is not possible. So we have to select a combination. If combination is one among \(C_1, C_2, \ldots, C_k\), then the result is proved. If it is not one among \(C_1, C_2, \ldots, C_k\), then another optimum combination can be formed from the given bids for given quantity \(q_k < q\). This is a contradiction, so there is no other combination, which is optimum for \(q\). Hence the result follows. All the above results are also applicable to ask in reverse auctions (by changing sign).

### 5.5 Algorithm for Assignment

In our system the bids or asks are submitted by the users till certain deadline. After the deadline the buyers, to whom allotment is to be carried out are selected. After selecting the buyers, the constraints are formulated as per IPO policy. In formulating constraints some quantity intervals submitted by buyers are deleted and in some cases its bounds are replaced. Then group constraints are formulated. The constraint formulation has been separated from the main algorithm. Then our algorithm is executed on the formulated problem. The algorithm is a general one and can be used for any forward/reverse auction system. After obtaining optimum solution our algorithm finds out the VCG Payoff. Our algorithm is based on the above results and works as follows

1. We start with the highest (lowest) available price. Then we determine the maximum quantity that is available at that price after taking into account all the constraints. If there are two bids (asks) with same price, then we combine them. The
price, quantity combination is added to assignment list. If the quantity allocated exceeds the required quantity we stop. If we get a combination that cannot be assigned to an earlier combination of bids, we create another new combination. We save its contribution value, quantity remaining (look ahead), combination list and buyer’s (seller’s) contribution in separate table structure.

(2) When we select next bid, which of the conditions 1-4 (Table-5.3) hold is tested. After verifying the results we form an optimum combination for quantity $q_i$ at any stage. If the allocated quantity exceeds the quantity required then we stop. Otherwise we add the price quantity points to appropriate combinations to obtain an assignment. We also eliminate the combinations, which are not likely to be in optimum solution.

(3) Repeat the above two steps till requirement is completely fulfilled.

(4) Then determine VCG Payoff for each buyer (seller). This is determined by subtracting the respective buyer’s (seller’s) contribution from the final solution.

We hereby state forward and reverse auction algorithms. The constraint formulation algorithm is not stated as it is a straightforward read and replace operation. Let A be the list of asks and B be the list of bids. The algorithms are as follows.

<table>
<thead>
<tr>
<th>Algorithm (1.1) forwauct /* Main Algorithm */</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Sort all bids on descending order of price within same price sort on descending order of quantity</td>
</tr>
<tr>
<td>2. while (there is an unassigned bid or unfulfilled demand) repeat steps 3 thru 9</td>
</tr>
<tr>
<td>3. Find out maximum quantity available for price $p_i$ after verifying capacity constraints on buyers and other group constraints</td>
</tr>
<tr>
<td>4. Verify whether optimality condition is satisfied. If verified exit</td>
</tr>
<tr>
<td>5. Add the price, quantity combination to appropriate assignment combination. Eliminate combinations which will not be in optimum solution.</td>
</tr>
<tr>
<td>6. Calculate and save the buyer’s contribution</td>
</tr>
<tr>
<td>7. Calculate the value of objective function</td>
</tr>
<tr>
<td>8 Mark the bid as assigned</td>
</tr>
<tr>
<td>9 Add quantity to demand fulfilled</td>
</tr>
<tr>
<td>10 Select the combination with the highest value.</td>
</tr>
<tr>
<td>11 Compute VCG Payoff</td>
</tr>
</tbody>
</table>

Figure. 5.1. Algorithm for Forward Auction

The algorithm for reverse auction is similar and is as follows.
The working of our algorithm is shown with the help of a simple example (Table 5.4).

**Theorem 5:** The algorithm always generates optimum solution for any quantity \( q \) if it exists.

**Proof:** At any stage our algorithm generates optimum solution for quantity \( q_i \). If quantity exceeds the required quantity, then the optimum solution is obtained and we stop. In other case we get an optimum solution for quantity \( q_i + q_j \) in next stage. Repeat this till assigned quantity exceeds quantity \( q \). This is the optimum solution. The algorithm will terminate when quantity \( q \) is assigned or all combinations are created and no bid is left out. At this stage we will have optimum solution for quantity \( q_k \). If \( q_k < q \), the demand is more and supply is less. So no optimum solution for quantity \( q \), from the existing bids exists.

Complexity: The complexity of our algorithm is always polynomial. There are \( n \) buyers and each buyer has submitted \( m \) points. In the best-case scenario, a combination can be created by scanning \( n \) bids. Each bid has \( m \) assignments. Scanning \( nm \) assignments in the worst case can create any combination. Then each combination can be tested for optimality in linear time. So the time complexity of our algorithm is \( O(nm) \). In each stage we scan \( n \) bids and one interval. In the worst case time complexity is \( O(nm) \) or polynomial. Once combination is obtained we need to scan them to obtain the
combination with the highest value. This can be done in linear time complexity. Apart from this sorting is the hardest part. Since sorting time complexity is of the order of $O(n \log n)$, the time complexity of this algorithm is $O(n^2) + O(nm \log nm)$ in worst case, which compares favorably with the time complexity order $O(n^3)$ presented in [KPS2003]. Additionally VCG Payoff gets computed in linear time complexity.

Example: Suppose that four buyers have submitted bids as follows. The output of our algorithm for different quantities is shown in column 5-9 of Table – 5.4. At each stage algorithm combination arrives at optimum solution. The algorithm has been implemented in C++. The data sets of asks and bids of different sizes were generated randomly. Each data set consisted of number of asks with ask price, quantity, ask size, bid size, bid price and bid quantity. Size of data sets varied from 5 to 1500. The results were compared with unconditional optimum solution and some solutions obtained with the help of MATLAB package. It can also be seen that time complexity of our algorithm is always polynomial. The figure – 5.1 indicates the comparative performance of proposed solution against algorithm proposed in [KPS2003]. It can be seen that the above algorithm works in case a few or all buyers submit price quantity pair instead of set of price quantity intervals.

<table>
<thead>
<tr>
<th>Buyer</th>
<th>Minimum Quantity</th>
<th>Maximum Quantity</th>
<th>Price</th>
<th>Quantity</th>
<th>Buyer</th>
<th>Quantity</th>
<th>Price</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>20</td>
<td>4</td>
<td>10</td>
<td>17</td>
<td>170</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>8</td>
<td>3</td>
<td>10</td>
<td>18</td>
<td>180</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>15</td>
<td>6</td>
<td>Total</td>
<td>350</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>20</td>
<td>5</td>
<td>27</td>
<td>4</td>
<td>10</td>
<td>17</td>
<td>170</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>3</td>
<td>10</td>
<td>18</td>
<td>180</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>7</td>
<td>15</td>
<td>2</td>
<td>15</td>
<td>7</td>
<td>105</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>12</td>
<td>Total</td>
<td>355</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>17</td>
<td>8</td>
<td>45</td>
<td>3</td>
<td>20</td>
<td>15</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>22</td>
<td>6</td>
<td>2</td>
<td>7</td>
<td>15</td>
<td>105</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>26</td>
<td>5</td>
<td>4</td>
<td>18</td>
<td>14</td>
<td>252</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>657</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>10</td>
<td>18</td>
<td>60</td>
<td>3</td>
<td>20</td>
<td>15</td>
<td>300</td>
</tr>
<tr>
<td>11</td>
<td>20</td>
<td>15</td>
<td>2</td>
<td>7</td>
<td>15</td>
<td>105</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>30</td>
<td>12</td>
<td>4</td>
<td>30</td>
<td>12</td>
<td>360</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>40</td>
<td>10</td>
<td>1</td>
<td>3</td>
<td>10</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>790</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>10</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>20</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>30</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>40</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
This case is treated as a single point being submitted by the buyer. So algorithm creates a combination by selecting price, quantity pair in appropriate combinations. In case a bid is all or nothing type of bid, then algorithm selects this combination, in case quantity remaining is higher than quantity q. If the remaining quantity does not exceed q, then the point cannot be selected as buyer’s constraint means that buyer requires entire quantity.

Figure. 5.3 Comparative performance of the algorithms

So in case there are bids with price higher than the bid with all or nothing constraint, then this bid (i.e. with all or nothing constraint) may not get selected. Our algorithm handles all these different cases.

5.6 Generalization of Multi Unit Auctions with Constraints
We now consider the case where all buyers submit bids as ordered pairs of prices and quantities. All bids are of type 3. In this case i\textsuperscript{th} buyer’s bid is \([(bp_{i1},bq_{i1}), v_i, (bp_{i2},bq_{i2}), (bp_{i3},bq_{i3}), \ldots, and (bp_{ik},bq_{ik})]\). It is interpreted as i\textsuperscript{th} buyer is willing to pay \(bp_{i1}\) per unit for bq\textsubscript{1} units, (\(bp_{i1}+bp_{i2}\)) per unit for (bq\textsubscript{1}+bq\textsubscript{2}) units, (\(bp_{i1}+bp_{i2}+bp_{i3}\)) per unit for
(bq_1 + bq_2 + qb_3) units and \( \sum_{j=1}^{k} bp_{ij} \) \( \sum_{j=1}^{k} bq_{ij} \) for \( \sum_{j=1}^{k} bq_{ij} \) units. The set \( v_i \) is the set of values of other attributes of the bid.

It can be seen that in this case the bid price represents the marginal cost that buyer is willing to pay for additional units. The main deviations, here from the case where buyer submit bids using marginal decreasing piecewise bidding language are as follows.

1. A buyer submits price quantity pair instead of a price and quantity interval
2. In optimum solution one or more price quantity pairs can be selected
3. Unlike in marginal decreasing piecewise bidding language, where price and quantities in different pairs may not always have well defined relationships. In marginal decreasing piecewise bidding language the price decreases and quantity increases.

Further when \( q_i = 1 \) for all \( i \), then this is a usual case of bids in multi unit auctions. This is the first deviation from usual multi unit auctions. In another deviation we consider capacity constraints and group constraints. In the absence of any type of constraint and when \( q_i = 1 \), our auction reduces to usual single object multi unit auction. Further buyer can specify whether his bid is all or nothing type. This bid is interpreted as buyer requires entire quantity \( \sum_{i=1}^{k} q_i \) and is not willing to accept anything lesser. The group constraints and capacity constraints are exactly same as those considered earlier. We can the rewrite above bid excluding other attribute set \( v_i \) as (it will be referred as price-quantity part of the bid).

\[ B_i^T = [(bp_{i1}, bq_{i1}), (bp_{i2}, bq_{i2}), (bp_{i3}, bq_{i3}), ..., (bp_{ik}, bq_{ik})] \]

where \( bp_{il} = \sum_{j=1}^{l} p_j \) and \( bq_{il} = \sum_{j=1}^{l} q_j \)

Suppose that \( n \) buyers (sellers) submit such bids (asks). Let us define

\[ x_{ij} = \begin{cases} 1 & \text{if } j \text{th element of } i \text{th bid is selected} \\ 0 & \text{otherwise} \end{cases} \]

\( i = 1, 2, \ldots, n \) and \( j = 1, 2, \ldots, K \).

\[ x_{ij} = 1 \quad \text{if } j \text{th element of } B_i^T \text{ is selected for the } i^{th} \text{ bid, } i = 1, 2, \ldots, n \text{ and } j = 1, 2, \ldots, K \]
If a buyer (seller) specifies only a single price, quantity pair then \( k = 1 \).

Let \( q_{ij} \) be the quantity sold to the \( i^{th} \) buyer at price \( b_{p_{ij}} \) (i.e. \( j^{th} \) element). Then we define our optimization problem as follows

\[
\text{max} \sum_{i=1}^{n} \sum_{j=1}^{K} x_{ij}q_{ij}b_{p_{ij}}
\]

The constraints are defined as follows

**Quantity Constraint:** Total quantity \( \sum_{j=1}^{K} q_{ij} \leq q, \quad i = 1, 2, ..., n, \quad j = 1, 2, ..., K \) \hspace{1cm} (5.27).

**Capacity Constraint:** Let \( b_{q_{i_{\text{max}}}} \) be the maximum quantity required by \( i^{th} \) buyer with \( i^{th} \). Then capacity constraint of buyer is

\[
\sum_{j=1}^{K} q_{ij} \leq b_{q_{i_{\text{max}}}}, \quad i = 1, ..., n \hspace{1cm} (5.28).
\]

If the buyer specifies \( b_{q_{i_{\text{min}}}} \) as minimum quantity that must be purchased by \( i^{th} \) buyer

\[
b_{q_{i_{\text{min}}}} \leq \sum_{j=1}^{K} q_{ij} \hspace{1cm} (5.29).
\]

In case \( i^{th} \) buyer specifies bid as all or nothing type of bid then it can be captured by the constraint

\[
\sum_{j=1}^{K} q_{ij} = \sum_{j=1}^{K} g_{ij} \hspace{1cm} (5.30).
\]

It may be noted that more than one buyer can submit such bids. It can be seen that if the buyer specifies minimum required quantity, then the quantity is required to be allocated to that buyer.

**Group Constraint:** Let \( A \) be the set of buyers, such that there is limitation on total quantity that can be purchased (sold) to them. There can be more than one such set. Let \( q_{a} \) be the quantity, then we have

\[
\sum_{i} \sum_{j=1}^{K} q_{ij} \leq q_{a}, \quad \text{where} \quad i \in A \hspace{1cm} (5.31).
\]

The other constraints are \( \sum_{j=1}^{K} x_{ij} \leq K, \quad x_{ij} = 0,1, \quad \text{and} \quad x_{ij-1} \leq x_{ij} \)

for all \( i = 1, 2, ..., n \) and \( j = 1, 2, ..., K \) and all \( b_{q_{ij}} \) are non negative integers. Further \( x_{i0} = 0 \) for all \( i \)

\[
q_{ij} \leq b_{q_{i_{\text{u}}}}
\]

where \( b_{q_{i_{\text{u}}}} \) is the upper limit of \( j^{th} \) pair submitted by \( i^{th} \) buyer. \hspace{1cm} (5.32).
It can be seen that this formulation is a particular case of formulation (5.1). The main difference is constraint (5.32). It ensures that buyer acquires $q_{i1}$ units before acquiring further $q_{i2}$ units. In this formulation a buyer can obtain different units at different prices, which is not possible in case of bids of type 2. Due to this, right hand side (RHS) of first constraint in (5.32) is $K$ here. In this case we can improve our earlier algorithm. The problem for reverse auctions can also be similarly defined. The formulation presented in [BVSV2002] is a particular case of this formulation. It is a case where $bq_{ij} = 1$ for $i$ and $j$, and group constraints are not present. Since quantity is not a decision variable the formulation in [BVSV2002] is linear. We use the same concept of contribution introduced earlier. So the contribution to the value of objective function at price $bp_i$ and quantity $q$ as $\text{cov}_{pi}(q)$ is

$$\text{cov}_{pi}(q) = bp_i q$$

(5.33)

The value of (5.33) is the maximum(minimum) for any $bp_i$, when $q$ is maximum (minimum). This helps in obtaining upper bound on improvements at any stage. If there is only one bid (ask) at price $bp_i$, with quantity $q$ and if there is no other bid with higher price or combination of bids of higher price, then this represents the optimum improvement. Otherwise optimum assignment is the combination of more than one bids(asks). Further such combination may be more than one price – quantity pair from any buyer.

Let the maximum price among the received bids be $bp_1$. Let $q_1$ be the maximum quantity that can be assigned at this price. Then it can be seen that the maximum possible improvement or upper bound on improvement at any stage is

$$\text{cov}_1 = bp_1 q_1$$

(5.34).

This is also the optimum assignment for quantity $q_1$.

Optimum assignment for a fixed quantity: In the current set up of multi unit auctions, a buyer can submit set of price quantity pairs. The price quantity part of buyer’s bid is of the form

$$B_i^T = [(bp_{i1}, bq_{i1}), (bp_{i2}, bq_{i2}), (bp_{i3}, bq_{i3}), ..., (bp_{ik}, bq_{ik})] \ i = 1, 2, ..., n$$

Combination: A set formed by selecting price-quantity pairs from different bids. A combination $C_i$ is of the form \{(bp_{i1},bq_{i1}),(bp_{i2},bq_{i2}), ..., (bp_{il},bq_{il})\}, $i = 1,2, ..., n$. 

210
Each pair \((bp_{ik}, bq_{ik})\) is a part of bid \(Bi^T\). It is \(k^{th}\) element of \(i^{th}\) bid. A combination can have more than one pair from any bid.

The total quantity \(q\) of the combination \(C_i\) is

\[
q = \sum_i \sum_j bq_{ij}
\]

The contribution to the value of objective function \(cov_{Ci}(q)\) from combination \(C_i\) is

\[
cov_{Ci}(q) = \sum_i \sum_j bp_{ij} bq_{ij}
\]

A combination \(C_i\) is optimum combination for quantity \(q\) if for any other combination \(C_j\), if and only if \(cov_{Ci}(q) \geq cov_{Cj}(q)\) for \(i \neq j\)

In order to obtain the optimum solution we determine the maximum price and the maximum quantity available at that price. We carry out assignment in stages, till the assignment of quantity \(q\) is completed.

Suppose that we are required to determine an optimum assignment for quantity \(q\). So first determine the highest price \(bp_1\) amongst all the bids. Determine the maximum quantity \(q_1\) that can be assigned as price \(bp_1\). If there are more than one bids with the highest price \(bp_1\), then we combine them and obtain the maximum quantity \(q_1\) that can be assigned. Suppose that there are \(i_1\) bids which have pair \((bp_1, bq_{ij})\), where \(i = 1, 2, \ldots, i_1, j = 1, 2, \ldots, K\). So our combination \(C_1\) is \{\((bp_1, bq_{i_1}), \ldots, (bp_1, bq_{i_1j})\}\). The quantity of the combination is

\[
q_1 = \sum_{i=1}^{i_1} \sum_{j=1}^{K} bq_{ij}
\]

Its contribution \(cov_{c1}(q) = bp_1 q_1\) \hspace{1cm} (5.35)

In the next stage determine the next highest price. Let \(bp_2\) be the next highest price. Then determine the maximum quantity \(q_2\) that can be assigned at this price. Suppose that there are \(i_2\) bids which have pair \((bp_2, bq_{ij})\), where \(i = 1, 2, \ldots, i_2, j = 1, 2, \ldots, K\). The quantity \(q_2\) satisfies the relation

\[
q_2 = \sum_{i=1}^{i_2} \sum_{j=1}^{K} bq_{ij}
\]

We add these points to combination \(C_1\). So our combination \(C_1\) is \{\((bp_1, bq_{i_1}), \ldots, (bp_1, bq_{i_1j}), (bp_2, bq_{ij}), \ldots, (bp_2, bq_{i_2j})\}\). The quantity of the combination is \(q_1 + q_2\).

Its contribution \(cov_{c1}(q) = bp_1 q_1 + bp_2 q_2\) \hspace{1cm} (5.36)
It can be easily verified that this combination is optimum for quantity $q_1 + q_2$. It can be seen that contribution of the combination can be improved if any one term in (5.36) can be improved. In this case both terms cannot be improved, so the combination is optimum for $q_1 + q_2$. The contribution can be improved only if there is a bid, which contains higher price quantity pair. In this case there is no such bid. The quantity assigned is also the maximum possible. As the second term cannot be improved, (5.36) is the highest possible improvement in this stage. So it is optimum assignment for quantity $q_1 + q_2$.

**Theorem 6:** Let $C_1 = \{(bp_1, bq_{1j}), ..., (bp_n, bq_{nj})\}$ be an optimum combination for quantity $q_{k-1}$. Let its contribution be $cov_{C1}(q_{k-1})$. Suppose that the following condition is satisfied.

1. For any bid $B_i$, where $i = 1, 2, ..., n$, there is no price $bp_m$ in any price quantity pair $(bp_i, bq_j) \not\in C_1$, such that $bp_m > bp_i$, for all $i$ and $bp_i \in (bp_i, bq_j) \in C_1$. i.e. price quantity pair in $C_1$. (In other words all price quantity pairs, which do not belong to $C_1$ have lower price.)

2. Let $bp_k$ be the price such that, $bp_k < bp_i$ for all $bp_i \in (bp_i, bq_j) \in C_1$ and $bp_k > bp_j$ for all $bp_i \in (bp_i, bq_j) \not\in C_1$. (In other words it is the highest price among the remaining price quantity pairs.) Let $bq_k$ be the maximum quantity that can be assigned at this price.

Then the combination $C_2 = \{(bp_1, bq_{1j}), ..., (bp_n, bq_{nj}), (bp_k, bq_k)\}$ is optimum for quantity $q_k$ ($= \sum_{i=1}^{k} bq_i$) and its contribution is $cov_{C2}(q_k)$.

**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 to the value of objective function of combination $C_2$ is

$$cov_{C2}(q_k) = cov_{C1}(q_{k-1}) + bp_k bq_k$$

The contribution can be improved, if and only if the second term can be improved. This is due to the fact that the first term cannot be improved by assumption. In this case it is not possible to improve the second term since its contribution $bp_k bq_k$ is the highest among the
remaining bids (and hence cannot be improved). In the same way, \( b_{q_k} \) is the maximum quantity that can be assigned. So the second term cannot be improved. Hence \( \text{cov}_{c_2}(q_k) \) is the maximum improvement in the value of objective function. So the result follows by induction.

This result helps us in determining optimum quantity at any stage. The remaining things remain similar to earlier setup hence they are not repeated here. The algorithm is a general one and can be used for any similar multi unit auction system. After obtaining optimum solution our algorithm finds out the VCG Payoff. Our algorithm is based on the above results and works as follows:

1. We start with the highest (lowest) available price. Then we determine the maximum quantity that is available at that price after taking into account all the constraints. If there are two bids (asks) with same price, then we combine them. The price, quantity combination is added to assignment list. If the quantity allocated exceeds the required quantity we stop.

2. Then select next highest price. Determine the maximum quantity that can be assigned at this price after taking into account all the constraints. If there are more than one bid which have this price in some price quantity pair, then combine them. If the quantity assigned exceeds the required quantity then stop. Otherwise repeat steps (1) and (2) till either all bids are assigned or the required quantity is obtained.

3. If all bids are assigned and required demand is not fulfilled, then no optimum assignment for that quantity exists.

4. If at any stage, in steps in (1) and (2) it is determined that entire quantity can not be assigned to all or nothing bid, then that bid is removed from optimum solution. Ties can be resolved in any predetermined ways.

We hereby state the algorithms for multi unit auction. The constraint formulation algorithm is not stated as it is a straight forward read and replace operation. Let \( AK \) be the list of asks and \( BD \) be the list of bids. The algorithm is as follows.
Algorithm (2) multiunit /* Main Algorithm */
1. Sort all bids on descending order of price within same price sort on descending order of quantity
2. while (there is an unassigned bid or unfulfilled demand) repeat steps 3 thru 9
3. Find out maximum quantity available for price $p_i$ after verifying capacity constraints on buyers and other group constraints
4. Verify whether optimality condition is satisfied. If verified exit
5. Add the price, quantity pair to the combination.
6. Calculate and save the buyer's contribution
7. Calculate the value of objective function
8. Mark the bid as assigned
9. Add quantity to demand_fulfilled
10. Select the combination with the highest value.
11. Compute VCG Payoff

Figure. 5.4. Algorithm for Multi Unit Auction

The working of our algorithm is shown with the help of a simple example (Table 5.5).

**Example:** Suppose that five buyers have submitted bids as shown in the first four columns of Table – 5.5. The first two buyers belong to a group, for which there is a group constraint. The fifth buyer has submitted a bid with all or nothing constraint. The group constraint is that only 12 units can be allocated to this group. There are totally 40 units. The output of our algorithm for different quantities is shown in column 6-9 of Table – 5.5. At each stage algorithm combination arrives at optimum solution.

**Theorem 7:** The algorithm always generates optimum solution for any quantity $q$ if it exists.

**Proof:** At any stage our algorithm generates optimum solution for quantity $q_i$. If quantity $q_i$ exceeds the required quantity, then the optimum solution is obtained and we stop. In other case we get an optimum solution for quantity $q_i + q_j$ in next stage. Repeat this till assigned quantity exceeds quantity $q$. This is the optimum solution. The algorithm will terminate when quantity $q$ is assigned or all combinations are created and no bid is left out. At this stage we have the optimum solution for quantity $q_k$. If $q_k < q$, the demand is more and supply is less. So no optimum solution for quantity $q$, from the existing bids exists.

**Complexity:** The complexity of our algorithm is always polynomial. There are $n$ buyers and each buyer has submitted $m$ points. In the best case scenario, a combination can be created by scanning $n$ bids. Each bid has $m$ points. So a combination can be created by
scanning nm points. Then each combination can be tested optimality in linear time. So the time complexity of our algorithm will be O(nm). In each stage we scan n bids and one interval. In the worst case, time complexity will be O(nm) or polynomial. Once combination is obtained we need to scan them to obtain the combination with the highest value. This can be done in linear time complexity. Apart from this, sorting is the hardest part. Since sorting time complexity is of the order of O(nm log nm), the time complexity of this algorithm is O(n²) + O(nm log nm) in worst case, which compares favorably with the time complexity order O(n³) presented in [KPS2003].

5.7 Determining the Payable Amount
In this section, the amount payable for each buyer or seller is determined. One option is to obtain the VCG Payoff of each buyer. As already stated, VCG mechanism, has a number of desirable properties, like incentive compatibility (IC) and efficiency (EE). Such properties may be desirable in IPO auctions, by means of which government owned enterprises or public resources are privatized.

Table 5.5 Bids received from Buyers

<table>
<thead>
<tr>
<th>Buyer</th>
<th>Price</th>
<th>Quantity</th>
<th>Cumulative Price</th>
<th>Quantity</th>
<th>Quantity Assigned</th>
<th>Price</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50</td>
<td>5</td>
<td>50</td>
<td>5</td>
<td>8</td>
<td>110</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>5</td>
<td>90</td>
<td>10</td>
<td>2</td>
<td>8</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>5</td>
<td>120</td>
<td>15</td>
<td>4</td>
<td>8</td>
<td>96</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>5</td>
<td>145</td>
<td>20</td>
<td>3</td>
<td>9</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>7</td>
<td>70</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>8</td>
<td>100</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>77</td>
<td>7</td>
<td>177</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>6</td>
<td>237</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>5</td>
<td>282</td>
<td>26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>4</td>
<td>314</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>90</td>
<td>9</td>
<td>90</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>54</td>
<td>7</td>
<td>144</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>3</td>
<td>165</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>96</td>
<td>8</td>
<td>96</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>6</td>
<td>156</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>5</td>
<td>201</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5*</td>
<td>110</td>
<td>8</td>
<td>110</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>7</td>
<td>180</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* All or nothing bid
It may be desirable that these properties are transferred to persons who value them most and truthful bidding is the dominant strategy.

Let $V_0$ be the optimum solution of the mixed integer programming problem formulated in (1). Then VCG Payoff of $i^{th}$ buyer $vcg_i$ is computed as follows

$$
vcg_i = V_0 - V_{oi} \quad i = 1, 2, 3, \ldots, n
$$

(5.37).

where $V_{oi}$ is the optimum solution of problem (5.1) obtained after removing $i^{th}$ buyer.

The term $vcg_i$ is referred as VCG discount of the $i^{th}$ buyer. The amount payable ($p_b_i$) by $i^{th}$ buyer is obtained as

$$
p_b_i = b_p_i b_q_i - vcg_i \quad i = 1, 2, 3, \ldots, n
$$

(5.38).

where $b_q_i$ is the quantity purchased by $i^{th}$ buyer and $b_p_i$ is the bid price. In the case of reverse auctions (i.e. sellers), VCG discount is obtained in a similar manner and is added to the product of ask amount and quantity sold. Since the nature of problem is symmetric with respect to buyer or seller, in further analysis from now we do not exactly state seller separately.

It can be seen that in order to obtain the amount payable for each buyer and seller, one is required to solve n optimization problems. The steps are

1. Remove a buyer (seller). There are (n-1) buyers now.
2. Obtain optimum solution of new optimization problem
4. Repeat steps (1) to (3) for each buyer.

Due to this computational complexity Vickrey auctions are not widely used. We now state an algorithm, which can obtain VCG Payoff of each buyer (seller) without solving set of optimization problems.

Optimum Solution of new problem: Suppose that the optimum solution of problem (5.1) has been obtained. In order to obtain VCG discount a new optimization problem excluding $k^{th}$ buyer is formulated. It can be seen from theorem (4) that it is always possible to obtain an optimum combination for any quantity $q$. Suppose that there is a combination $C_i$. It contains an assignment $(b_{p_k}, b_{q_k})$ for $k^{th}$ buyer. Let $valc_i(q)$ be the value of the combination. After removing $k^{th}$ buyer the value of combination changes to $valc_i(q) - b_{p_k} b_{q_k}$.
In order to obtain optimum combination of new problem, we go to stage before selecting \(k^{th}\) buyer’s bid. Suppose that at this stage there are \(l\) optimum combinations for different quantities. If any of these combinations have bids from \(k^{th}\) buyer go to stage before that. Then optimum combination can be obtained by finding out optimum combination for remaining quantity and then selecting combination with the maximum quantity. The algorithm can be stated as

```
Algorithm (3) vcgcal
While there is buyer with payoff not calculated do {
    Remove \(l^{th}\) buyer;
    While there is no new combination {
        Remove \(l^{th}\) buyer’s bid from combination;
        Add combination to combination list ;}
    While there is no new combination {
        Obtain optimum combination for remaining quantity ;
        Find out optimum combination ;
        Find payoff of buyer ;
        Set payoff calculated to true ;
    }
}
```

Figure. 5.5. Algorithm for VCG payoff

This algorithm generates optimum solution for set of optimization problem without actually solving the optimization problem. In this algorithm, initially the optimum combination excluding \(l^{th}\) buyer is found out. This combination is then combined with remaining combinations without \(l^{th}\) buyer to obtain optimum combinations.

**Theorem 8**: The algorithm always generates optimum solution if it exists.

**Proof**: Let \(C_i\) be the optimum combination for original optimization problem. Let \(q\) be the total quantity of this combination. In this combination at some stage \(l^{th}\) buyer’s bid is added. Let \(C_l\) be the optimum combination before selecting \(l^{th}\) buyer’s bid. This is an optimum combination for quantity \(q_l\). Let \(bq_l\) be the quantity of \(l^{th}\) buyer. After adding \(l^{th}\) buyer’s quantity, the combination is optimum for quantity \(q_l + bq_l\). So we consider all such optimum combinations before selecting \(l^{th}\) buyer. These combinations are optimum for respective quantities. These combinations are independent of \(l^{th}\) buyer. All these bids are selected because they have higher contribution then \(l^{th}\) buyer’s bid. Hence they are optimum even for new optimization problem. In the next stage by determining the optimum combinations for remaining quantity and combining it with the combinations
before selecting \( l \)\(^{th} \) buyer’s bid the optimum combination for new optimization problem can be obtained. Our algorithm exactly follows this procedure and hence obtains optimum solution for new optimization problem, if it exists.

Suppose that the assignments \((b_{p_1}, b_{q_1}), (b_{p_2}, b_{q_2}), \ldots, (b_{p_k}, b_{q_k})\) , replace the assignment \((b_{p_l}, b_{q_l})\). Then change in the value of a combination due to this replacement is

\[
bp_{l}bq_{l} - \sum_{i \neq l} bp_{i}bq_{i}.
\]

Complexity of Algorithm: The time complexity of this algorithm is always polynomial but in the best case the solution can be obtained with linear time complexity. It can be seen that in the worst case scenario an optimum solution can be obtained by scanning all created combinations. Then this procedure is repeated for all the buyers. Suppose that before \( l \)\(^{th} \) buyer is selected there are \( n_l \) combinations. The optimum combination for the remaining quantity can always be created with polynomial time complexity. However the time complexity can be improved if combinations before including \( l \)\(^{th} \) buyer and after \( l \)\(^{th} \) buyer are stored. At any stage an optimum combination can be created by scanning these combinations.

In case of single object multi unit auctions with constraints, there is a possibility that after deleting a bid or an ask, the resulting problem may not have a feasible solution. If this happens, then it is not possible to obtain an optimum assignment. In this case the amount payable by a buyer or a seller cannot be obtained. As such we generalize Uniform Price Auction mechanism for IPO and multi unit auctions with constraints scenario. As earlier, let BD be the set of all bids. It is assumed that all buyers have their private valuations. Let \( b_{v_i} \) be the private valuations of the \( i \)\(^{th} \) buyer. Suppose that buyer pays \( b_{a_{p_i}} \) per unit of quantity and acquires \( b_{q_i} \) units. The utility of the \( i \)\(^{th} \) buyer is \( b_{u_i} = (b_{v_i} - b_{a_{p_i}})b_{q_i} \).

Generalized Uniform Price Mechanism (GUP): Divide set BD into \( k \) mutually exclusive sets \( B_{g_1}, B_{g_2}, \ldots, B_{g_k} \), satisfying following

\[
\text{(a)} \quad BD = B_{g_1} \cup B_{g_2} \cup B_{g_3} \cup \ldots \cup B_{g_k}
\]

and \( B_{g_i} \cap B_{g_j} = \phi \) for all \( i \neq j \).

Each set \( B_{g_i} \), represents the set of buyers in \( i \)\(^{th} \) group. It can be seen that a buyer can be in only one of the sets \( B_{g_1}, B_{g_2}, \ldots, B_{g_k} \). A buyer can be an employee, individual or an enterprise. Depending upon the type of buyer he is classified into one of the sets. Let \( b_{q_i} \) be the upper limit on the quantity that can be allocated to group \( B_{g_i} \). In the next step,
the optimum assignment is determined using the earlier algorithm. Let \( B_{oi} \) be the set of selected bids from \( i^{th} \) group by the algorithm. Then the clearing price is determined. In the straight forward generalization of uniform price auction there are two possibilities

(1) use uniform price for all groups, so all buyers irrespective of group pay the same price

(2) use different clearing price for different groups, but each buyer in the group pays the same price.

Let \( pc_i \) be the clearing price for \( i^{th} \) group. Let \( B/l \) be the set loosing (i.e. the set of bids which are not selected) bids from all groups. Let \( bp_{ij} \) be bid price of \( i^{th} \) selected buyer in \( j^{th} \) group \( B_{oj} \).

Let \( bpc \) be the price such that \( bp_{ij} \geq bpc \) for all \( i \) and \( j \), and for any bid \( B_m \) in \( B/l \), with bid price \( bp_m \), such that \( bp_m > bpc \), there are at least one \( i \) and \( j \), such that \( bp_{ij} < bp_m \).

Then in the first case, the clearing price for \( i^{th} \) group as

\[ pc_i = bpc. \]

We set the same clearing price for all groups.

In the second case, the clearing price for the \( i^{th} \) group is set at the highest loosing bids in that group. In our mechanism we set the clearing price for \( j^{th} \) group as

\[ pc_j = bpc, \]

where \( bpc \) is the price of the highest loosing bid in \( B/l \) such that,

\[ bpc \leq bp_{ij} \text{ for all } i. \]

In other words the clearing price for \( i^{th} \) group is set to the price of the highest loosing bid, across the groups, such that it does not exceed bid price of selected bids in that group.

We call our mechanism as generalized uniform price mechanism (GUP). The mechanism for the seller can be similarly defined. Then we show that this mechanism is individually rational, efficient and strategy proof.

**Theorem 9:** Generalized uniform price (GUP) mechanism satisfies the IR property.

**Proof:** It can be seen that in GUP, if buyers or sellers are not selected then they do not pay or receive any amount. So they have 0 utility. In case buyers or sellers participate in auction then

(3) The per unit price to be paid by any buyer in \( i^{th} \) group is \( pc_i \). This price satisfies the relation \( pc_i \leq bp_{ij} \) (if \( j^{th} \) buyer’s bid is selected in \( i^{th} \) group).
(4) In the same way the seller never receives the per unit amount, which is lesser than per unit amount of ask submitted by him. The per unit price to be received by a seller is always at least as much as his ask price.

It may be noted that per unit price paid by each buyer in auction does not exceed per unit bid amount. In the same way per unit price to be received by each seller in auction is not lower than per unit price submitted. So all the participants have nonnegative gains or utility after auction clears. Due to this, the GUP mechanism is always IR.

**Theorem 10**: If prices and volumes of the buyers (or sellers) are public information GUP mechanism is strategy proof.

**Proof**: Suppose that the selected bids are distributed in k groups $B_{g1}$, $B_{g2}$, ..., $B_{g_k}$. Let $p_{ci}$ be the clearing price of $i^{th}$ group. As this proof is stated for $i^{th}$ group, the subscript $i$ is omitted from this proof. Let $bv_j$ be the private valuation of the $j^{th}$ buyer and $bp_j$ be the bid price submitted. Let $bq_j$ be the quantity purchased by the $j^{th}$ buyer. Suppose that $j^{th}$ buyer’s bid is in $i^{th}$ group, with $p_{ci}$ as clearing price. The utility $bu_j$ of $j^{th}$ buyer is

$$bu_j = (bv_j - p_{ci}) bq_j .$$

(a) Bidding higher than true valuation: Suppose that buyer’s valuation $bv_j \geq p_{ci}$ and buyer bids $bp_j$. The buyer’s utility is

$$bu_j = (bv_j - p_{ci}) bq_j , \quad \text{if } bp_j \geq p_{ci}$$

$$= 0 \quad \text{otherwise}$$

It can be seen that buyer cannot improve his utility by bidding $bp_j > bv_j$. In this case the buyer’s bid is selected buy his utility does not improve. This happens as the amount payable does not depend upon buyer’s bid. In other words his utility by bidding higher remains same as utility, he gets if he bids $bv_j$.

(b) Bidding lower than true valuation: Suppose that buyer’s valuation $bv_j \geq p_{ci}$ and buyer bids $bp_j$ which satisfies $bp_j < bv_j$. The buyer’s utility is

$$bu_j = (bv_j - p_{ci}) bq_j , \quad \text{if } bv_j > bp_j \geq p_{ci}$$

$$= 0 \quad \text{if } bv_j \geq p_{ci} > bp_j$$

It buyer bids lower than his true valuation, buyer’s bid is selected only if $bp_j \geq p_{ci}$. In case $bp_j < p_{ci}$ buyer’s bid is not selected. In the first case buyer’s utility does not improve by bidding lower amount. In the second case buyer’s bid is not selected, even though his valuation is higher than the clearing price. The utility of buyer is 0 in this case. So by
bidding lower than his true valuation buyer does not improve his utility and in some cases may have 0 utility.

(c) Suppose that $b_{vj} < pc_i$, i.e. buyer’s valuation is smaller than clearing price. Suppose that buyer bids $b_{pj}$. The utility of the buyer is

$$b_{u} = \begin{cases} 
(b_{vj} - pc_i) q_{b_{j}}, & \text{if } b_{pj} \geq pc_i \\
0, & \text{otherwise}
\end{cases}$$

In case buyer bids $b_{pj} > b_{vj}$, buyer’s bid can get selected if $b_{pj} \geq pc_i$. However in this case buyer has negative utility. In case buyer bids $b_{pj} < b_{vj}$, this bid is not selected and he has 0 utility.

Combining arguments in (a), (b) and (c), it can be seen that buyer does not improve his utility by bidding untruthfully.

(d) Further it can be seen that by changing his bid price, buyer’s bid will not get selected in different group. This is due to the fact that a bid can be in only one group and he is already included in $i^{th}$ group. So buyer cannot improve his utility by being included in a group, where clearing price is less than $pc_i$. The buyer’s bid is included in the group, depending upon his type. In order to get included in different group, buyer needs to change his type. It is independent of the buyer’s price. By bidding lower amount, buyer may not be included in a group with lower clearing price than $pc_i$. So lower bidding will not improve his utility.

We can extend this argument to seller’s case to show that the seller’s utility does not improve by reporting higher or lower price than his valuation.

This proves that GUP mechanism is strategy proof. It may be noted that at any point our algorithm selects a bid with the highest contribution for given quantity. The contribution is the highest for any fixed quantity, when the corresponding price is the highest. So in GUP mechanism, the bids of only those buyers are selected, who value them the most. So the mechanism is efficient.

5.8 Misreporting of Volumes

It can be seen that in GUP mechanism, a buyer or seller does not have incentive to misreport his price. A buyer does not have incentive to report higher volumes. However in some cases buyers or sellers can misreport quantities and this can affect the clearing price. For example a buyer can decrease and report his volume. In some cases this can
bring down the clearing price for a group. On the other hand, seller can misreport his volumes to improve his utility. In some cases buyers or sellers can misreport their respective volumes to improve utility. Consider the following scenario

Suppose that in \( j^{th} \) group, the lowest winning bid is at \( i = m_i \). If \( i^{th} \) buyer reduces his demand then, \((m_i - 1)^{th}\) or subsequent bid is selected either partly or fully. This happens as demand is reduced. This reduces the clearing price and buyer can improve his utility.

In other words, clearing price has been pushed down. In some cases sellers can misreport the volumes. If seller can misreport volume, it can push up market clearing price. However such manipulations are quite difficult in our set up. We propose the following modification to GUP mechanism to minimize manipulations by buyers and sellers. In order to achieve it, an attempt is made to minimize incentive to misreport volumes. However the scheme requires that there are minimum two buyers or sellers in any group.

Let there be \( m_i \) buyers in \( i^{th} \) group \( B_{gi} \).

Let \( q_i \) be the total quantity sold or purchased in \( i^{th} \) group \( B_{gi} \).

Let \( bq_{ij} \) be quantity purchased by \( j^{th} \) buyer in \( i^{th} \) group \( B_{gi} \).

It is easy to verify \( \sum_{j=1}^{m_i} bq_{ij} = q_i \) for all \( i = 1, 2, ..., k \).

Let \( bq_o_{ji} \) be the quantity purchased by all other buyers except \( j^{th} \) buyer in group \( B_{gi} \).

So \( bq_o_{ji} = q_i - bq_{ij} \).

Let \( b_{pi_{(min)}} \) be the minimum price of the selected bid prices in the \( i^{th} \) group. We also have

\[ b_{pi_{(min)}} \geq pc_i \] for \( i = 1, 2, ..., k \).

Let us define \( bin_{oi} = (b_{pi_{(min)}} - pc_i) \) for all \( i = 1, 2, ..., k \)

It can be easily verified that \( bin_{oi} \geq 0 \).

In this case our mechanism works in the same way as GUP, however there is no uniform clearing price for any group. In this case if \( j^{th} \) buyer’s bid is selected in \( i^{th} \) group the per unit payable amount for this buyer is

\[ bpua_{ij} = pc_i + \left( \frac{bq_o_{ji}}{q} \right) bin_{oi} \] for \( i = 1, 2, ..., k \), \( j = 1, 2, ..., m_i \) \hspace{1cm} (5.39).

So if \( bq_{ij} \) is the total quantity purchased by \( j^{th} \) buyer in \( i^{th} \) group, then the total amount payable is

\[ bpa_{ji} = bpua_{ji} \cdot bq_{ij} \].
In the similar way amount receivable by different sellers can be defined.
Let there be \( n_i \) sellers in \( i^{th} \) group \( B_{gi} \).
Let \( q_i \) be the total quantity sold in \( i^{th} \) group \( B_{gi} \).
Let \( aq_{ij} \) be quantity sold by \( j^{th} \) seller in \( i^{th} \) group \( B_{gi} \).
It is easy to verify \( \sum_{j=1}^{n_i} aq_{ij} = q_i \) for all \( i = 1, 2, ..., k \).

Let \( aq_{oji} \) be the quantity purchased by all other sellers except \( j^{th} \) seller in group \( B_{gi} \).
So \( aq_{oji} = q_i - aq_{ij} \).

Let \( api_{(max)} \) be the maximum price of the selected ask prices in the \( i^{th} \) group. We also have
\[
api_{(max)} \leq pc_i \quad \text{for} \quad i = 1, 2, ..., k
\]
Let us define \( ain_{oi} = (pc_i - api_{(max)}) \) for all \( i = 1, 2, ..., k \)
It can be easily verified that \( ain_{oi} \geq 0 \). As \( api_{(max)} \geq pc_i \).

Similarly per unit price to be received by \( j^{th} \) seller in \( i^{th} \) group is
\[
srua_{ji} = \left( pc_i - \frac{aq_{oji}}{q} \right) ain_{oi}
\]

In this scheme each buyer or seller who participates in the auction pays and receives different per unit price, unlike in GUP. This scheme is referred to as Discriminatory Multi Unit Auction mechanism (DMA).

**Theorem 11:** DMA mechanism satisfies the IR property.

**Proof:** It can be seen that in DMA as in GUP, if buyers and sellers are not selected then they do not pay and receive any amount. So they have 0 utility. In case buyers or sellers participate in auction then
In DMA per unit price paid by any buyer in \( i^{th} \) group
\[
bpua_{ji} = pc_i + \left( \frac{bq_{oji}}{q} \right) bin_{oi}
\]

It can be easily verified that \( bpua_{ji} \leq bp_i \) for \( i = 1, 2, ..., m_i \).

This follows as
\[
\left( \frac{bq_{oji}}{q} \right) < 1 \quad \text{as} \quad q_i = bq_{oji} + bq_{ij} \quad \text{(by definition) and} \quad bq_{ij} > 0.
\]

Multiplying both sides by \( bin_{oi} \) we get
Adding $pc_i$ to both sides we get

$$pc_i + \left( \frac{b_{go}}{q} \right) bin_{oi} \leq pc_i + bin_{oi}.$$ 

Using definition of $bin_{oi}$ we get

$$pc_i + \left( \frac{b_{go}}{q} \right) bin_{oi} \leq bpi_{(min)}.$$ 

It may be noted that $j^{th}$ buyer’s bid is selected in $i^{th}$ group if and only if $bp_j \geq bpi_{(min)}$. So the price paid by any buyer is bounded by $bpi_{(min)}$.

This proves the result that all the participating buyers have nonnegative gains from participation.

Exactly similar argument can be extended to show that all sellers have nonnegative gains from participation.

As, per unit price paid by each buyer in auction does not exceed the bid amount and the same to be received by each seller in auction is not lower than the price received, all participants have positive utility or have nonnegative gains after auction clears. Due to this, DMA mechanism is always IR.

**Theorem 12:** DMA mechanism satisfies incentive compatibility property. Further if any buyer (seller) decreases (increases) the quantity, he pays (receives) more (less) price respectively per unit. The mechanism is also false name proof and efficient.

**Proof:** It can be easily verified that the amount paid by any buyer or seller is independent of his bid/ask price. It depends upon the bid prices and ask prices submitted by others. Further, as in GUP mechanism, a buyer or seller cannot improve his gain by bidding higher or lower. The utility of $j^{th}$ buyer in $i^{th}$ group for a single unit of object is

$$bu_{ij} = bv_{ij} - pc_i \text{ for } i = 1, 2, \ldots, k, \ j = 1, 2, \ldots, m_i$$

If buyer’s valuation $bv_{ij} \geq pc_i$, then buyer’s utility does not increase by bidding higher amount, as the amount to be paid does not change. He may lose the auction by bidding lower amount. On the other hand if buyer’s valuation $bv_{ij} < pc_i$, and if buyer bids higher amount, buyer may win but has $bu_{ij} < 0$. In case he bids lower amount, he has 0 utility. So the mechanism is incentive compatible.
Suppose that $j$\textsuperscript{th} buyer in $i$\textsuperscript{th} group requires quantity $b_{ij}$. Suppose that instead of $b_{ij}$ buyer submits his demand as $bnq_{ij} < b_{ij}$.

Let $q_i$ be the total quantity sold in $i$\textsuperscript{th} group, if $j$\textsuperscript{th} buyer submits his true demand. Let $qn_i$ be the demand when buyer submits $bnq_{ij}$. Then we have the following relations,

\[ q_i = b_{qo} + b_{ij}, \]
\[ qn_i = b_{qo} + bnq_{ij}, \quad \text{since} \quad bnq_{ij} < b_{ij}, \quad \text{so} \quad q_i > qn_i, \]
\[ \frac{1}{q_i} < \frac{1}{qn_i}, \quad \text{so we get} \]
\[ \left( \frac{b_{qo}}{q_i} \right) < \left( \frac{b_{qo}}{qn_i} \right) \]

This proves the result that if a buyer reduces the quantity required he pays more amount per unit of quantity. This is true even if, another bid is selected on account of demand reduction. This happens as there is no change in $q_i$ and quantity purchased by others increases. It can be seen that in this case

\[ q_i = qn_i \text{ and } bnq_{ij} > b_{ij}. \]

So the per unit payable amount increases.

Conflict and false name proof bids: It can be further seen that there is a conflict between the different buyers or sellers in the sense that, if any buyer or seller decreases his volume and the volumes of others remain unchanged, then others gain at the cost of this buyer. This conflict occurs in case there is no change in the quantity purchased. This can be easily verified.

Let $d = b_{ij} - bn_{ij}$. So for any other buyer say $k$, the factor $\left( \frac{b_{qo}}{q} \right)$, changes to

\[ \left( \frac{b_{qo} - d}{q - d} \right). \]

It may be noted that $\left( \frac{b_{qo} - d}{q - d} \right) \leq \left( \frac{b_{qo}}{q} \right)$.

So all other participants gain. So buyer pays more by decreasing his demand and at the same time other buyers gain because they pay lesser price. Due to this conflict no buyer or seller has incentive to reduce his volume. In some cases there may be another bid, which gets selected due to demand reduction. Suppose that there is no change in total quantity $q_i$. So the factor $\left( \frac{b_{qo}}{q} \right)$ remains unchanged for $k$\textsuperscript{th} buyer ($k \neq j$). Due to this utilities of others remain unaffected.
In electronic auctions bids are submitted remotely. It is possible that buyers may submit false bids. Such false bids are bids submitted under different identification just to improve utility. Such bids are submitted under fictitious names [YSM2000]. It can be seen that in DMA mechanism a buyer does not have incentive to submit false name bids. Suppose that \( k^{th} \) buyer submits false name bids, on account of these bids, there is increase in demand say by \( d \) units.

In case these bids are not selected there is no effect on clearing price. Suppose these bids get selected, then \( k^{th} \) buyer pays more price. This is due to the fact that the factor \( \left( \frac{bq_o}{q} \right) \), changes to \( \left( \frac{bq - u + d}{q + d} \right) \). It may be noted that \( \left( \frac{bq - u + d}{q + d} \right) \geq \left( \frac{bq - u}{q_r} \right) \). In fact by submitting false name bids a buyer may increase the per unit price to be paid by others in some cases. So DMA does not have incentive for submission of false name bids. It can be seen that at the same time utility of some others remain unaffected. There are other possibilities like the quantity allocated to \( j^{th} \) buyer decreases, whereas there is no change in other allocation. Suppose that false name bid is selected and quantity \( q_i \) is allocated to it. However total quantity \( q_t \) remains unchanged. In other words false name bid replaces a bid selected earlier either fully or partly. In any case this does not affect total volume of other buyers but can decrease the quantity allocated to \( j^{th} \) buyer. The quantity allocated to \( j^{th} \) buyer does not increase. In some cases it may decrease. Due to this buyer cannot improve his utility. The bids selected in this mechanism are same as in GUP mechanism. So by following similar argument as in GUP mechanism it can be proved that this mechanism is efficient.

Exactly similar argument can be followed to show that this mechanism is incentive compatible for sellers and seller receives lesser price per unit in case he reports higher quantity.

The percentage gain for a buyer (as his demand decreases) was worked out from 100 randomly generated data sets. Each data set was randomly generated and it consisted of bid prices, quantity and values for other attributes. The effect on percentage gain as demand decreases can be seen in figure 5.6. At the same time effect of false name bids on his percentage gain can be seen in figure 5.7.
5.9 Conclusion
In this chapter we have formulated the problem of IPO as single object multi unit auctions with different types of constraints. Our formulation is a general one and some of the multi unit auction formulations studied earlier are particular cases of our formulation. Then we develop an algorithm to obtain optimum solution. After obtaining optimum solution, an algorithm to obtain VCG Payoff, without solving set of optimization problem, has been presented. Then we propose a design of strategy proof mechanism GUP for single object multi unit auctions with different types of constraints. It is shown that our mechanism is strategy proof, individual rational and efficient. Then we propose design of a mechanism DMA, which is efficient, strategy proof and individually rational. These properties ensure that truthful bidding is the dominant strategy, which is a very important property in electronic auctions. We further show that our mechanism is false name proof and reduces incentive for buyer to report decrease in volume.

Figure. 5.6- Effect on gain due to change in demand. It indicates how gain of buyer decreases in case he decreases his volume
Figure. 5.7 – Effect of False Name Bids. How the gain of buyer decrease by submitting false name bids