THE DISTRIBUTION GAME:
A DISTRIBUTION POLICY SIMULATOR

by

Peter L. Jackson
John A. Muckstadt
Russell Rushmeier
Purpose of the Game

The Distribution Game is an educational game designed to interest students in the problems of ordering and allocating stock in multi-echelon distribution systems. The focus of the game is on developing rules for placing orders and for allocating stock to different locations in the system. The Game illustrates the economic tradeoff between order costs and inventory costs and demonstrates the need to provide safety stock to protect against stochastic variability in customer demand. The Game can also be used to illustrate the role of information in managing distribution systems.

The Game can be played with little prior exposure to inventory theory. However, it is intended to be used in conjunction with lecture material that reviews formal models of single- and multi-echelon inventory control. At a minimum, students should be familiar with the Economic Order Quantity model and some model that explains the relationship between order points and safety stock. These concepts are reviewed below in the section entitled Basics.

The Game was originally developed by Russell Rushmeier, Jack Muckstadt, and Peter Jackson in the School of Operations Research and Industrial Engineering, Cornell University, Ithaca, N. Y. This version of the Game has been programmed by Peter Jackson and incorporates suggestions by Christian Delporte and Marc Lambrecht.

Game Features

The Distribution Game is an animated simulation of a simple two-echelon distribution system consisting of one central warehouse and three regional distributors, or retailers. Only one product is stocked and sold. The player is responsible for four daily decisions: how much, if any, additional stock to order from the supplier and how much, if any, stock to ship from the central warehouse to each of the three retailers. The goal of the game is to maximize cumulative net profit. Revenue comes from sales made by the retailers. If the retailers are out of stock then either sales are lost or customers are willing to wait for backordered demand. The Game instructor determines whether to use the lost sales or the backorder version. Costs are
of three types: cost of goods sold, which is proportional to the number of units sold, order costs, which are proportional to the number of orders placed or shipments sent, and inventory holding costs, which are proportional to amount of inventory on hand or in transit. Note that holding costs are charged even for inventory in transit from the supplier to the central warehouse.

The player interacts with the Game using tree-structured menus. Each menu includes a help key that displays an explanation of the menu. Figure 1 shows the menu structure and the Appendix lists the menu explanations.

The Game is designed for use in competitions. Each simulation reproduces the same sequence of pseudo-random customer demands so that differences in reported net profit between players are due entirely to the different game strategies the players employ. The random number seed is hidden from the players. Access to view the random number seed and to change any of the game parameters is protected by a password. The instructor can change the password as well as the random number seed and other game parameters and store the result on disk. The password is encoded prior to being stored on disk.

The Game also features an auto-play mode in which an internal algorithm, called the "Helper," is used for placing orders and making shipments. The Helper attempts to optimize the order cost-holding cost tradeoff and maintain a balanced distribution of stock among the retailers. No claim is made concerning the optimality of its decisions: it is provided only to increase the level of competition. The auto-play feature can be used as an additional player in game competitions. The Helper is described in detail below.

To illustrate the value of information, the Game can be played in a mode that hides information concerning retailer stock levels from the player. Players are unlikely to achieve as high a profit in this version of the Game as they do in the full information version.
Figure 1. The Distribution Game Menu Structure
The Game is written using TURBO PASCAL version 2.0 from Borland. A planned later release of the Game will be compiled using TURBO PASCAL version 4.0. This later release will allow the instructor to replace the Helper with alternative algorithms.

Two language versions of the Game are currently available: English and French. The menus are in English but the game screen, menu explanations, and parameter names are stored as ordinary ASCII text files on disk. English and French versions of these files now exist. These files can be revised or translated by the instructor using any ASCII file text editor.

Equipment Requirements

The Distribution Game runs on an IBM PC or compatible under PC-DOS 3.0 and requires 128K RAM and either the monochrome display adapter or the color graphics display adapter. When the simulation terminates the Game provides the option of displaying a time series plot of several statistics collected during the game play. This plotting option requires the color graphics display adapter but is not essential to game play. Hence the Game can be played on almost any PC.

Game Play

To play the Game, first boot your PC using PC-DOS 3.0 or higher (refer to the Operator’s Manual of your PC for an explanation of this procedure). Insert the Game diskette in the drive labelled A. Type the following commands, following each command with the Enter or carriage return key on your keyboard:

a:

dist

This will initiate the program. After the credits screen is displayed, you will see the game screen displayed. The bottom two lines of the screen summarize the function keys that are active. Hit function key F1 to display an explanation of the Game. Hit function key F2 to step through this explanation and return to the main menu.
The Game is controlled by means of function keys and menus. An explanation of any menu can be obtained by hitting function key F1 when that menu is displayed.

To leave the Game program, hit function key F2 repeatedly, until the menu labelled "Exit Program" is active and then hit function key F10.

The menu explanations are reproduced in the Appendix of this manual. Note that the menu labelled "Report Statistics" is available only after the end of the Game has been reached.

Screen Printouts
A feature of PC-DOS enables the player to obtain a printout of any screen in the Game. Hit the key combination <Shift> <PrtSc> to obtain such a printout. Some symbols used on the screen, such as those used in drawing lines and boxes, may not be reproduced correctly on the printer unless the printer conforms to the original IBM standard for the PC.

All but one of the screen displays in the Game are text screens. The Cumulative Plot is a graphics screen. It is not possible to obtain a printout of this screen unless a special print driver has been loaded into the PC's memory prior to running the Game program. Such a driver is provided for the IBM dot matrix printer as part of the PC-DOS operating software. To load the driver, place the PC-DOS diskette in the drive labelled A and enter the command "graphics", without the quotes. If the operation is successful, it should display the message "Resident part of GRAPHICS installed." After this, place the diskette with the Distribution Game in the drive labelled A and enter the command "dist", without the quotes. Game play proceeds as described in the previous section. To obtain a screen printout of the Cumulative Plot hit the key combination <Shift> <PrtSc> when the plot is fully displayed.
Finite Horizon

The Game ends after a fixed number of days. This is essential for game play but players are quick to note that the strategies that are appropriate in the real world, that is, long term operating strategies, are sub-optimal as the end of the Game approaches. Consequently, players will tend to end the Game with almost no inventory in the system. One could imagine a Game version that gives some sort of credit for ending inventory in order to encourage a more long run view. However, the better such a credit scheme is at capturing the long run value of ending inventory, the more complicated it would be to explain to players. In the interest of simplicity, therefore, no credit is given in the Game for ending inventory.

Suggestions for Competitive Play

It is recommended that, prior to a game competition, the instructor change the random number seed and the password and then save the new parameter set on disk. This can be done using the Edit Parameters menu (the password is "cosmos"). The students can then be assigned to teams (no more than three students per team for optimal interaction) and each team given a diskette containing the Game program ("dist.com"), the language version file ("dist.000"), the help files ("*.hlp"; that is, all files with the extension ".hlp"), and the saved parameter file ("dist.001").

After a demonstration and practice session with the software, instruct the teams to recall the set of saved parameters using function key F4 in the Select Screen Type and Game Version Menu (F6 from the Main Menu). Allow the teams at least an hour to play the Game. Announce that the teams may play the Game as many times as they wish during this period: they need only submit their best score. They can interrupt game play at any time and reset to starting conditions by means of function key F6 on the main menu. Request a screen printout of their best final score and a screen printout of the corresponding Report Statistics screen. Any screen printout that contains the word "Modified" must be disallowed. Use these printouts to identify the winning team and award them a prize.
The topics listed in the section below entitled "Basics" can be used as a basis for a discussion of the Game after the competition. Encourage the winning team to present a description of their strategy. Compare their results and approach with that of the Helper. If time permits, run a second competition. For the second competition, handicap the winning team from the first competition by requiring that they play Game Version 2. Alternatively, allow the students to play a "Just-In-Time" version of the Game as described below.

It might also be instructive to obtain a screen printout of the Cumulative Plot from each team. Obtain one such printout from their first play of the Game and one from the their last play of the Game to demonstrate the learning that has taken place. See the section above entitled "Screen Printouts" for important instructions concerning printouts of this plot.

**Text Translation**

The file "dist.000" is an ordinary ASCII text file containing all the text used by the program except for the menus and the help files. If this file is missing, then the program automatically displays the English version of all text. The instructor can translate this file into any other language that uses the ASCII character set by using a text file editor. Be careful to first make a backup copy of the original file for reference purposes. Also be careful to translate the file "line for line": the Game program expects a fixed number of lines in a specific sequence. The help files can also be translated but it is not necessary to translate these files line for line. The Game program does not require any particular format for the help files.
Basics

The following topics constitute a basic set of principles that can be illustrated using the Distribution Game. Depending on the background of the students, the instructor may wish to introduce some of the topics prior to having the students play the Game. Other topics, such as fair share allocation, are best discussed after the students have had the opportunity to play the Game. The presentation of these topics here is at a level intended for second-year college students, those with at least some exposure to calculus and elementary probability theory. The references list several textbooks that provide an expanded treatment of several of these topics.

Economic Order Quantity Model

The Economic Order Quantity Model is a model of the tradeoff that arises between fixed order costs on the one hand and inventory holding costs on the other when determining order sizes. The Economic Order Quantity is the order size that optimizes this tradeoff.

If the rate of demand is constant and the transit or lead time required to receive delivery of an order is constant then orders can be timed so that delivery occurs precisely when on hand inventory is about to run out. In this situation the average amount of inventory on hand between the placement of two orders is equal to half the order size. Let \( Q \) denote the order size. If each order is of size \( Q \) and orders are timed to just prevent run-out but no earlier, then average inventory will equal \( Q/2 \) for as long as the system operates in this manner. Let \( C \) denote the purchase price of the stock, per unit. The average inventory investment amounts to \( CQ/2 \). Assuming annual inventory holding costs are proportional to average inventory investment, let \( I \) denote the holding cost rate per dollar invested per year. Annual inventory holding cost is therefore given by \( ICQ/2 \).

If \( Q \) is the order size and \( D \) is the annual demand rate, then the average number of orders that will be placed in one year is \( D/Q \). Let \( K \) denote the cost of placing an order. Average annual ordering cost is given
by KD/Q.

Total average annual cost is given by the sum of average annual ordering cost and average annual inventory holding cost: KD/Q + ICQ/2. This cost function describes the tradeoff involved in the choice of order size. For large Q, inventory holding cost is high; for small Q, ordering cost is high. The minimum of this function is found by elementary calculus: the optimal order size, called the Economic Order Quantity (EOQ), is given by EOQ = (2KD/(IC))^{0.5}.

For example, consider the problem of choosing an order size in the Game. Referring to the list of parameters for the standard version of the Game, observe that K = $200 per order and C = $70 per unit. The annual demand rate, D, is found by multiplying the number of retailers times the average daily demand rate per retailer times the number of days in a year. Since the daily demand is uniformly distributed over the values {0, 1, 2, 3, 4}, the average daily demand rate is 2 units per day. Hence, the total average annual demand rate for the system is given by D = 3x2x200, or D = 1200. Two holding cost parameters are listed: one for the warehouse and one for the retailers. In deciding how much stock to bring into the system with each order, the appropriate parameter is the one for warehouse. The tradeoff between the holding cost at the warehouse and the holding cost at the retailers can be examined when determining the size of shipments to make to the retailers. Hence, I = 0.21. Substituting these parameters into the formula for the Economic Order Quantity yields that EOQ = (2x200x1200/ (0.21x70))^{0.5} = 181. At a daily demand rate of 6 units, orders would be placed on average every 30 days, or, roughly, once a month.

Considering the problem of choosing good shipment sizes to make to the retailers, the appropriate values of the parameters are: K = $2.75, C = $70, D = 400, and I = 0.04. Note that the appropriate holding cost factor is the additional holding cost entailed in holding stock at the retailer rather than at the warehouse. The flow of stock into the system is controlled by the placement of orders with the supplier. Therefore, the warehouse order size
current inventory position is equal to or less than \( R \). Suppose current inventory position is exactly equal to \( R \). If an order is placed now, then when the order is delivered a lead time from now net inventory (that is on hand inventory net of backorders) will be equal to \( R \) less the demand that occurs over the duration of the lead time. This latter quantity is referred to as lead time demand. Let \( D_L \) denote the lead time demand. If the demand rate were constant then it would be possible to set the order point exactly equal to this lead time demand so that the retailer would run out of stock precisely when the delivery arrives. If the demand rate is stochastic then there is a probability that lead time demand will exceed \( R \) and result in a stockout. Let \( \text{Prob} \{ D_L > R \} \) denote the probability of a stockout and let \( \text{SO}(R) \) denote the expected number of shortages during the replenishment lead time given that inventory position is exactly equal to \( R \) at the beginning of the replenishment period. The number of order cycles per year is approximately \( D/Q \) so the expected number of shortages per year is approximately \( D/Q \cdot \text{SO}(R) \). Let \( P \) denote the per unit cost of a stockout. The expected annual shortage cost is thus \( P \cdot D/Q \cdot \text{SO}(R) \).

On the other hand, if demand is stochastic there is also a probability that there will be stock on hand when the delivery arrives. Let \( \text{SS}(R) \) denote the expected number of units on hand at the end of a replenishment lead time given that inventory position is exactly equal to \( R \) at the beginning of the replenishment period. This amount is referred to as safety stock. In expectation, this amount is always in the system. Consequently, the annual cost of holding this safety stock is \( \text{IC} \cdot \text{SS}(R) \).

The expected annual cost affected by the order point is \( P \cdot D/Q \cdot \text{SO}(R) + \text{IC} \cdot \text{SS}(R) \). Increasing the order point, \( R \), increases inventory holding costs by increasing the safety stock but reduces the stockout cost by reducing the expected number of stockouts per cycle. The optimal order point is the value of \( R \) that equates the marginal increase in holding costs with the marginal decrease in stockout costs. Using calculus, the optimal order point is the value of \( R \) such that \( P \cdot D/Q \cdot \text{Prob} \{ D_L > R \} = IC \cdot \text{Prob} \{ D_L \leq R \} \), or, rewritten, the value of \( R \) such that \( \text{Prob} \{ D_L > R \} = IC (PD + ICQ) \). The quantity on
basically determines the average amount of inventory in the system. The retailer shipment size determines primarily how this inventory is divided between the warehouse and the retailer. Applying the EOQ formula yields that the economic shipment size is \( EOQ = 28 \). The average demand rate is 2 units per day, so a shipment size of 28 means that the warehouse will make a shipment to the retailer every 14 days, or, roughly, twice a month. The standard parameters of the Game were chosen so that retailer shipments would occur more frequently than warehouse orders. The instructor can change these parameters to permit the students to explore other situations.

Several assumptions of the Economic Order Quantity model do not apply in the Game: demand does not occur at a constant rate but occurs stochastically, and the goal is to achieve the maximum net profit after a fixed number of days, not to achieve minimum long run average cost. Consequently, a fixed order size of 180 units ordered every 30 days and a fixed shipment size of 28 units every 14 days are unlikely to constitute an optimal policy. Nevertheless, the EOQ formula provides a starting point for developing good ordering and shipment policies for the Game.

**Order Points**

Another key idea that is illustrated naturally in the Game is the need to plan for transportation lead times. If these transit times were zero then the player could wait until inventory is just about to run out and then instantly replenish it. With lead times of several days, the player must plan ahead, anticipate stockouts and place orders or make shipments several days in advance. It is common to use order point rules: place an order or make a shipment when some measure of inventory reaches a predetermined critical value, or order point. The typical inventory measure at the retailer level is called inventory position: on hand inventory plus stock in transit less backordered demands.

Without developing an order point model in full here, it is possible to summarize some essential insights. Let \( R \) denote the reorder point used to decide when to replenish stock. That is, a replenishment order is placed if
the right hand side could be referred to as the **economic stockout probability**. As a heuristic, we use the same model for both the lost sales and the backorder case.

The above argument assumes that inventory position exactly equals \( R \) at the time a replenishment order is placed. As players of the Game will note, however, the lumpiness of demand is such that inventory position often falls below a fixed order point before an order can be placed. This can be thought of as a consequence of the length of the review period. If the player could observe the system in real-time, watching customers arrive one by one, then replenishment orders could be placed precisely when inventory position equals \( R \), assuming no two customers arrive at exactly the same time. As it is, the player only observes the system at discrete points in time, at the review times. The length of the review period in the Game is one day and the instructor can increase this length to emphasize this problem. Let \( V \) denote the length of the review period and let \( D_{V+L} \) denote the demand that occurs during one review period plus one replenishment lead time. If inventory position equals \( R+1 \) then no order will not be placed for at least one review period. Consequently, inventory position at the time of the next delivery will be less than or equal to \( R+1-D_{V+L} \). A more conservative estimate of the stockout probability is therefore given by

\[
\text{Prob}\{D_{V+L} > R+1\} \leq \text{ICQ}/(PD+ICQ).
\]

More accurate methods of determining near-optimal order point, order quantity combinations can be found in Porteus [1985].

The problem of deciding when to ship stock from the warehouse to the retailers can be approached as an order point problem. In this case the lead time is five days from the warehouse to the retailer and the review period is one day. The unit cost of a stockout is the opportunity cost of a lost sale: \$100 revenue less \$70 purchase price, i.e. \( P = \$30 \). From the discussion of order quantities above, \( Q = 28 \) is a reasonable value. In this case the
appropriate value of the inventory holding rate is \( I = 0.26 \), the full cost of holding stock at the retailers. The economic stockout probability is given by 
\[
\text{ICQ)/(PD+ICQ) = 0.26 \cdot 70 \cdot 28/(30 \cdot 400+0.26 \cdot 70 \cdot 28) = 0.04074.
\]

The probability distribution for a single day's demand is given by the uniform distribution over the values \( \{0,1,2,3,4\} \). Convoluting this distribution six times yields the probability distribution for the total demand over 6 days. The values of the stockout probability \( \text{Prob}\{D_{V+L} > R+1\} \) for this distribution are tabulated in Table 1. From this table it can be seen that the smallest value of \( R \) satisfying \( \text{Prob}\{D_{V+L} > R+1\} \leq 0.04074 \) is \( R = 17 \). Consequently, making shipments to a retailer whenever its inventory position equals or falls below 17 is likely to provide a good tradeoff between lost sales on the one hand and inventory holding costs on the other.

As with the Economic Order Quantity model, there are many assumptions of this Order Point model that do not apply in the Game. Nevertheless, computing the economic stockout probability is a useful beginning to setting a good order point.
according to demand pattern, the concept is generalized to mean an allocation that equalizes the probability of a stockout at each retailer after a fixed interval of time. Where cost patterns differ by retailer, the concept can be further generalized to mean an allocation that equalizes cost-weighted stockout probabilities.

The fair share allocation is illustrated in Table 2 by means of five examples. For each example, the table lists the stock at each retailer immediately prior to allocation, the total amount of stock to allocate, called the stock budget, the total at each retailer of the stock prior to allocation plus the stock allocated, and whether alternative fair share allocations exist. Assume no stock is in transit at the time of the allocation. Example (a) illustrates a common case in which the stock levels after allocation are perfectly equalized. Examples (b) and (c) illustrate a case in which the requirement that stock allocations be integers prevents perfect equalization. In this case three equivalent fair share allocations are possible, only two of which are shown. Examples (d) and (e) illustrate that the fair share allocation to a retailer can be zero if the stock budget is small relative to the imbalance in the stock levels prior to allocation. Retailer 3 receives nothing in the fair share allocation of these latter two examples.

<table>
<thead>
<tr>
<th>Example</th>
<th>Retailer Stock Before Allocation</th>
<th>Stock Budget</th>
<th>Retailer Stock After Allocation</th>
<th>Alternate Solutions Possible</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>a</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>30</td>
</tr>
<tr>
<td>b</td>
<td>14</td>
<td>15</td>
<td>15</td>
<td>30</td>
</tr>
<tr>
<td>c</td>
<td>14</td>
<td>15</td>
<td>15</td>
<td>30</td>
</tr>
<tr>
<td>d</td>
<td>13</td>
<td>14</td>
<td>18</td>
<td>5</td>
</tr>
<tr>
<td>e</td>
<td>13</td>
<td>14</td>
<td>18</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 2. Examples of Fair Share Allocation

It is important to distinguish between stock that is allocated to a retailer and stock that is actually shipped to the retailer. Stock can be allocated to a retailer but held in reserve at the warehouse. At a later
date, this stock can be shipped to the retailer or it can be reallocated at that time to a retailer that is in greater need. This is a useful concept in deciding when to place orders with the supplier: all warehouse stock, including stock in transit to the warehouse, can be allocated to the retailers regardless of whether it is feasible or desirable to ship the allocations. The allocated stock levels can then be checked to see whether any retailer is in danger of a stockout over the lead time required to receive stock from the supplier. That becomes the basis for recognizing when it is time to place an order with the supplier. An application of this idea is described below in the section explaining the Helper.

A brief survey of some of the research literature concerning fair share allocation can be found in the article by Jackson and Wuckstadt [1988].

Game Version 2 and the Value of Information

It is evident from the discussion of fair share allocation that if a warehouse manager is going to attempt to keep stock levels at subsidiary retailers in balance it is necessary to have current information on the stock levels of all the retailers simultaneously. However, many multi-echelon distribution systems are controlled by means of simple policies that use only order points and order quantities. The retailers place orders when stock levels fall below their order points and these orders are the only data that the warehouse manager has concerning retailer stock levels. This makes the task of effective stock allocation very difficult. The likely result is that the manager must maintain more stock in the system in order to prevent stockouts at the retailers. The lesson is that more complete information together with the technique of fair share allocation can be used to achieve better performance. The improvement in performance must be weighed against the cost of implementing more extensive information systems and more sophisticated control policies. The tradeoff, therefore, is between investment in inventory to prevent stockouts and investment in information systems to achieve more effective stock distribution.
Economic Cost vs. Accounting Cost

The Game features an economic model of a distribution system. The costs and benefits that are relevant to the decisions of ordering and allocating stock have been collected and combined into a single performance measure, called here Net Profit but more aptly called Relevant Net Profit. Irrelevant costs and benefits have been excluded. In practice, some of these relevant costs are not easily measured or identified within standard accounting records. Order costs are typically not calculated on a per order basis but appear in more aggregated form as wages, salaries and freight charges. Inventory holding costs appear in different forms: lease costs for warehouse space, inventory insurance, and interest on corporate debt. It is easy to see therefore, that reports and performance measures constructed along traditional financial accounting lines can miss focusing attention on the tradeoffs that the Game is meant to highlight.

Real World Considerations

The Game is not intended to be a training simulator, exposing potential warehouse managers to important crises they may face in operating their warehouses. Rather, it is designed, in part, to interest students in formal approaches to repetitive decision-making situations. The Game is sufficiently complicated to defy closed-form analytical solutions but sufficiently repetitive in nature so that the application of a few basic principles yields a good solution. It is intended as a supplement to textbook material on single- and multi-echelon inventory theory. Once the basic principles have been identified, it is healthy to extend the discussion to more practical considerations: non-identical retailers, non-stationary, correlated and lumpy demand patterns, non-constant lead times, multi-item orders and shipments, and so on.
To illustrate the importance of information in operating multi-level distribution systems, Game Version 2 hides information concerning retailer stock levels from the player. Retailer stock levels are only displayed if they fall below a specified "order point". (The instructor can change this order point parameter.) Thus, this version mimics the situation of a retailer that provides no information to the warehouse until stock levels are low, whereafter he calls the warehouse daily to ask the status of his order. This lack of information will make it difficult for the player to achieve as high a profit in this version of the Game as was possible in Game Version 1.

The Just-In-Time Philosophy

An advocate of the Just-In-Time philosophy, e.g. Hall [1983], could argue that the Distribution Game focuses attention on the wrong problem. That is, the effort expended in devising and implementing good operating strategies would be better spent eliminating the root causes of the control problems. Shigeo Shingo [1985], for example, stresses that relatively simple techniques and only nominal investments are required in many cases to dramatically reduce setup times in manufacturing systems. By analogy, the fixed order costs taken as given in models of distribution systems may also be opportunities for cost reduction programs. In the absence of fixed order costs, the concern for judicious lot-sizing disappears.

The instructor can use the Distribution Game to illustrate such Just-In-Time concepts. By setting the fixed order cost and the fixed shipment cost to zero, using the Edit Parameters menu, the instructor can create a "Just-In-Time Distribution Game." There is a significantly different flavor to game play in this environment: the player becomes concerned with keeping the pipeline full of stock and responding day by day to changes in demand. Although the lot sizing problem disappears in this environment the lead times still force the player to be concerned with order points and allocation problems.
An Explanation of the Helper

This section explains the algorithm for ordering and allocating stock referred to as the Helper within the Game. No claim is made concerning its optimality: only that it should provide a reasonable level of competition with most players.

Notation

The algorithm is explained for the lost sales case. For the backorder case, references to retailer on hand inventory need to be replaced by references to retailer on hand inventory less backorders.

Let \( i = 0,1,2,3 \), index the four storage locations: the warehouse \((i = 0)\) and the three retailers. Let \( I_i \) denote inventory position at location \( i \), defined to be on hand inventory plus stock in transit to that location. Let \( V_0 \) denote warehouse on hand inventory. Let \( A_i(B) \) denote a fair share allocation of stock budget \( B \) to retailer \( i \). Figure 2 describes an algorithm for computing the vector \( A(B) = (A_1(B), A_2(B), A_3(B)) \).

1. Sort and relabel the retailers so that \( I_1 \leq I_2 \leq I_3 \).
2. Set \( A_i = 0 \), \( i = 1,2,3 \); set \( i = 1 \).
3. WHILE \( B > A_1 + A_2 + A_3 \) DO

BEGIN
4. Set \( U = B - A_1 - A_2 - A_3 \).
5. IF \( i < 3 \) AND IF \( (I_{i+1} - I_i) \cdot i < U \)

THEN FOR \( j = 1 \) TO \( i \) set \( A_j = I_{i+1} - I_j \);
ELSE FOR \( j = 1 \) TO \( i \) set \( A_j = A_j + U/i \).
6. Set \( i = i + 1 \).

END.

Figure 2. Fair Share Allocation Algorithm
Helper Order Points

Let $R_0$ denote the order point for the warehouse (called the supplier order point in the Game parameter list) and let $R$ denote the retailer order point. A decision is made to ship from the warehouse to retailer $i$ if two conditions are met: $A_i(V_0) > 0$ and $I_i \leq R$, that is, if there is any stock to ship to retailer $i$ under the fair share allocation of all the warehouse stock on hand and if the inventory position at retailer $i$ is at or below the retailer order point.

A decision is made to order more stock from the supplier if for any retailer $i$ the following condition is met: $I_i + A_i(I_0) \leq R_0$, that is, if the sum of inventory position at retailer $i$ plus the fair share allocation of all warehouse stock, including stock in transit, to retailer $i$ is at or below the warehouse order point.

Selection of the values of the order points $R_0$ and $R$ is left to the instructor by means of the Edit Parameters menu. A simple method for choosing $R$ is to use the economic stockout probability as described above. A simple method for choosing $R_0$ is to add $R$ to the expected retailer demand over the supplier-to-warehouse lead time. More precise methods are the subject of current research.

Helper Shipment Quantities

If the conditions to make a shipment from the warehouse to retailer $i$ are satisfied then the Helper proceeds to determine the appropriate shipment size. If $A_i(V_0) + I_i \leq R$, then the entire allocation $A_i(V_0)$ is shipped. Otherwise, let $W = A_i(V_0) + I_i - R$; this portion of the allocation need not be shipped all at once. If the demand rate were constant, this portion would be sufficient to last $W/D$ days, where $D$ is the expected daily demand rate at the retailer. Let $T$ denote the planning interval: $T = W/D$, in days. If $W$ is shipped in $n$ equal-sized, equally-spaced shipments and the demand rate is constant, then the total order and holding cost over the planning interval $T$ is given by $C(n) = Kn + HD(T/n)^2n/2$, where $K$ is the retailer order cost.
Conclusion

Video game style educational software has been found to be an effective means of capturing students' interest in problems of material logistics. The Distribution Game is an attempt to exploit this medium in the area of multi-echelon inventory theory. Several instructors have used the Game with good results. If you care to share your experiences with the Game, please forward your comments to Peter L. Jackson, Associate Professor, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, N.Y. 14853.
and $H$ is holding cost rate per unit per day. Let $n^*$ minimize $C(n)$ over the set of positive integers. The value $n^*$ is typically small and easily found using marginal analysis. Given $n^*$, the shipment size for retailer $i$ is $R - I_i + TD/n^*$.

One advantage of this approach is that it can be adapted easily to the finite horizon nature of the Game. In particular, if only $M$ days remain in the play of the Game, then the Helper takes $T = \min\{W/D, M-L\}$, where $L$ denotes the warehouse-to-retailer lead time.

**Helper Order Quantities**

If the conditions for a supplier order to be placed are met then the Helper uses the following procedure to compute the order quantity. Let $U_i = I_i + A_i(I_0) - R_0$. If $U_i < 0$, then $-U_i$ is referred to as a shortfall; otherwise $U_i$ is referred to as extra stock. Let $U^-$ denote the total shortfall over all the retailers and let $U^+$ denote the total extra stock. $U^-$ is the minimum order size. Let $D_0$ denote the total expected system demand rate per day and let $L_0$ denote the supplier-to-warehouse lead time. Suppose $M$ days remain in the play of the Game. Let $W_0 = (M-L_0-L)D_0 - U^+$. The maximum order size is given by $W_0 + U^-$. The planning horizon is taken to be $T_0 = W_0/D_0$.

Let $K_0$ denote the warehouse order cost and let $H_0$ denote the warehouse holding cost per unit per day. For $n_0$ equal-sized, equally-spaced orders the total order and holding cost for the planning interval $T_0$ is given by $C_0(n_0) = K_0n_0 + H_0D_0(T_0/n_0)^2n_0/2$, assuming a constant rate of demand. Let $n_0$ minimize $C_0(n_0)$ over the set of positive integers.

The Helper order size is taken to be $U^- + W_0/n_0^*$. However, if the total order size multiplied by the difference between the selling and purchase price is less than $K_0$ then the order is cancelled.
REFERENCES


THE DISTRIBUTION GAME

A COSMOS Comet

Concept by: P.L. Jackson, J.A. Muckstadt, and R. Rushmeier
Program by: Peter L. Jackson
Language Version: English

THE DISTRIBUTION GAME

DAY: 0

Sales $ 0.00
C.G.S. 0.00
Order Cost 0.00
Holding Cost 0.00
Net Profit $ 0.00

Main Menu:
F3 View | F4 Play | F5 Reset | F1 Help | F2 Quit

A-1
APPENDIX A

Screen Printouts from the Distribution Game
The game program is menu-driven. The menus are displayed on the last two lines of the screen. Special messages appear from time to time on the third last line of the screen.

The menus indicate which function keys are active and what each function key does. You have already discovered that function key F1 in the main menu gives you these instructions.

The other function keys in the main menu are:

F2 Quit. Leave the program.

F3 View. See a listing of the game parameters. The help key on the view menu provides a detailed explanation of these parameters.

F4 Play. Play the game. The help key on the play menu explains how to make orders and shipments.

F6 Reset. Reset the game to begin again. The reset menu allows you to specify other screen modes and game variations.

Pause. Hit F2 when ready.  

Credits

The Distribution Game was originally created by Peter Jackson, Jack Muckstadt, and Russell Rushmeier of the School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York.

Peter Jackson and Jack Muckstadt are the architects of COSMOS: The Cornell Simulator of Manufacturing OperationS, a graphical modelling system for exploring design and operational issues in manufacturing systems. "A COSMOS comet" refers to any of their more light-hearted projects.

This version has been programmed by Peter Jackson. It has benefited substantially from suggestions by Christian Delporte and Marc Lambrecht.

The French language version of the instructions was contributed by Claudine Henaux.

Pause. Hit F2 when ready.  

A-3
THE DISTRIBUTION GAME

Welcome to the Distribution Game. This is an educational game designed to interest you in the problems of ordering and allocating stock in multi-level distribution systems.

There are two stocking levels in this distribution system:
(1) the WAREHOUSE, and
(2) the RETAILERS.

As owner of the company, you control both levels. You decide:
(1) when to order, and how much to order from your supplier, and
(2) when to ship, and how much to ship to each retailer.

Customers buy the stock from the retailers and your object is to make as much money as possible from these sales.

"You can't sell from an empty truck,"
so make sure to keep the retailers supplied with stock.

Don't place orders or make shipments too frequently or your order costs will be too high. But, don't keep too much inventory in the system or your interest and holding costs will be too high.

Pause. Hit F2 when ready.

Screen Layout

Here is where you place orders with the supplier.

Sales $                   Here is where you make shipments to the retailers
C.G.S.                   180 << This box shows how much stock
Order Cost               the warehouse has.
Holding Cost
Net Profit $              << Make this big!

These boxes show how much stock each retailer has.

Pause. Hit F2 when ready.

Total customer demand is shown below each box.

F2 Go on
Number of days in game: 200
Review period: 1
Supplier to warehouse transit time (days): 15
Warehouse to retailer transit time (days): 5
Minimum daily demand per retailer: 0
Maximum daily demand per retailer: 4
Initial random number: Hidden
Will customers wait for backordered demands? No
Initial warehouse inventory: 180
Initial retailer inventory: 20
Selling price (dollars): 100.00
Purchase price (dollars): 70.00
Supplier to warehouse fixed order cost (dollars): 200.00
Warehouse to retailer fixed order cost (dollars): 2.75
Warehouse holding cost rate ($ / purchase $ / year): 0.21
Retailer holding cost rate ($ / purchase $ / year): 0.25
Number of days in year: 200
Retailer order point (Game 2): 20
Retailer order point (Game 1 Helper): 17
Supplier order point (Game 1 Helper): 46

View parameters.
F3 Edit | | | | | | F1 Help F2 Return

View Game Parameters

Number of days in game

The game is over when you reach the ending day of the game. To play again, hit the Reset key in the Main menu.

Review period

The review period is the number of days between when you are allowed to place orders. When you hit the 60 key in the Play menu, the game advances automatically to the next review day. If the review period is one day, then the 60 key advances the game to the next day.

Supplier to warehouse transit time (days)
Warehouse to retailer transit time (days)

Transit time refers to the number of days it takes for an order to travel from the supplier to the warehouse or from the warehouse to any one of the retailers.

Pause. Hit F2 when ready. | | | | | | | | F2 Go on

A-4
Selling price (dollars)

Sales = Number of units sold \times Selling price

Purchase price (dollars)

Cost of goods sold = Number of units sold \times Purchase price

Supplier to warehouse fixed order cost (dollars)
Warehouse to retailer fixed order cost (dollars)

Fixed order costs reflect such things as the value of the time required to process an order and all its associated paperwork, and the inspection and physical handling of the stock when it arrives.

Order costs = Number of supplier orders \times Warehouse fixed order cost
+ Number of retailer shipments \times Retailer fixed order cost

Pause. Hit F2 when ready. | | | | | | | | F2 Go on

Warehouse holding cost rate ($ / purchase $ / year)
Retailer holding cost rate ($ / purchase $ / year)
Number of days in year

Holding costs reflect such things as interest expense on the investment in inventory and storage expenses such as space rental. Warehouses are typically cheaper places to store inventory than retailers. This is reflected in a lower warehouse holding cost rate.

Holding cost for one day = Warehouse inventory (including orders in transit) \times Purchase price
\times Annual warehouse holding cost rate
/ Number of days in year
+ Total retailer inventory (including shipments in transit) \times Purchase price
\times Annual retailer holding cost rate
/ Number of days in year

Pause. Hit F2 when ready. | | | | | | | | F2 Go on

A-6
Minimum daily demand per retailer
Maximum daily demand per retailer

Customer demand on each day at a retailer is a pseudo-random number uniformly distributed over the range from the minimum to the maximum daily demand. Average daily demand = 0.5 * (maximum + minimum).

Initial random number

This number determines the sequence of customer demands. Each time you hit the Reset key on the Main menu, you reset the sequence of customer demands according to this number. To get a different sequence, you must change this number. See the Edit Parameters menu.

Pause. Hit F2 when ready. | | | | | | | | F2 Go on

Will customers wait for backordered demands?

If NO then if a retailer is out of stock it cannot make any sales.
If YES then retailer inventories are allowed to be negative to represent backordered demands. In this case, the game reports the "On time fill rate": the percentage of customer demands that are satisfied without being backordered. Think of your "score" in the game as Net profit * fill rate. The better your fill rate, the better your score.

Initial warehouse inventory
Initial retailer inventory

The game begins with a certain amount of inventory at each of the four locations: the warehouse and the three retailers.

Pause. Hit F2 when ready. | | | | | | | | F2 Go on

A-5
To change any of the displayed game parameters, hit

F3 Edit Parameters.

You will be prompted to supply a password in order to reach the Edit Parameters menu. Type the word "cosmos" (without the quotes) followed by the carriage return or enter key on your keyboard.

Your instructor may change the password.

Entering the password means that you are no longer playing the standard game. The game display will thereafter have the word "MODIFIED" on it. This is to inhibit the misuse of the Edit Parameters option in game competitions. The only way to remove the "MODIFIED" label is to quit the program and reload it.

To return to the Main menu, hit

F2 Return.

Pause. Hit F2 when ready.

| | | | | | F2 Go on

---

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of days in game</td>
<td>200</td>
</tr>
<tr>
<td>Review period</td>
<td>1</td>
</tr>
<tr>
<td>Supplier to warehouse transit time (days)</td>
<td>15</td>
</tr>
<tr>
<td>Warehouse to retailer transit time (days)</td>
<td>5</td>
</tr>
<tr>
<td>Minimum daily demand per retailer</td>
<td>0</td>
</tr>
<tr>
<td>Maximum daily demand per retailer</td>
<td>4</td>
</tr>
<tr>
<td>Initial random number</td>
<td>1949</td>
</tr>
<tr>
<td>Will customers wait for backordered demands?</td>
<td>No</td>
</tr>
<tr>
<td>Initial warehouse inventory</td>
<td>180</td>
</tr>
<tr>
<td>Initial retailer inventory</td>
<td>20</td>
</tr>
<tr>
<td>Selling price (dollars)</td>
<td>100.00</td>
</tr>
<tr>
<td>Purchase price (dollars)</td>
<td>70.00</td>
</tr>
<tr>
<td>Supplier to warehouse fixed order cost (dollars)</td>
<td>200.00</td>
</tr>
<tr>
<td>Warehouse to retailer fixed order cost (dollars)</td>
<td>2.75</td>
</tr>
<tr>
<td>Warehouse holding cost rate ($ / purchase $ / year)</td>
<td>0.21</td>
</tr>
<tr>
<td>Retailer holding cost rate ($ / purchase $ / year)</td>
<td>0.25</td>
</tr>
<tr>
<td>Number of days in year</td>
<td>200</td>
</tr>
<tr>
<td>Retailer order point (Game 2)</td>
<td>20</td>
</tr>
<tr>
<td>Retailer order point (Game 1 Helper)</td>
<td>17</td>
</tr>
<tr>
<td>Supplier order point (Game 1 Helper)</td>
<td>46</td>
</tr>
</tbody>
</table>

Edit parameters. Use cursor controls to select.

F3 Chng - | F4 Chng + | F6 Pswrd | F8 Save | F1 Help | F2 Return

A-8
Retailer order point (Game 2)

Retailers don't place orders in either version of the game.
You, as owner of the system, make all the decisions. Game version 2, however, is an attempt to mimic the situation in which your information system does not let you see everything that goes on. In a decentralized system, for example, the warehouse manager only sees orders from the retailers, not the day-to-day customer demands. To mimic this, game version 2 suppresses information about retailer inventory levels until these levels drop below the retailer order point.

To select game version 2, see the Reset menu.

Pause. Hit F2 when ready. | | | | | | | F2 Go on

Retailer order point (Game 1 Helper)

For players who have no one to compete against, a helper is included to suggest decisions in Game 1. The helper is based on a set of rules of when to order and how much to order. The helper attempts some optimization but cannot guarantee that its suggested decisions are the best possible. The purpose of the game is to encourage experimentation. Can you discover a better set of rules?

The helper uses these order point parameters to decide when to place orders and make shipments. You can affect the performance of the helper by changing these order points.

The helper is not available in the standard version of the game. To make the helper available, you must gain access to the Edit Parameters menu. See the next screen....

Pause. Hit F2 when ready. | | | | | | | F2 Go on
THE DISTRIBUTION GAME  (MODIFIED)  DAY:  0

Sales  $  0.00
C.G.S.  0.00
Order Cost  0.00
Holding Cost  0.00
Net Profit  $  0.00

Edit shipments. Use tab key to select.
F3 Help F4 Go | F6 Auto | F1 Help | F2 Return

Play Menu

You must make four decisions each day of the game:

- how much to order from the supplier, and
- how much to ship from the warehouse to each of three retailers.

At the beginning of each day, each of these decisions is reset to zero.

To select a decision, move the cursor until the decision is highlighted. Use the tab key or the cursor control keys on your keyboard to move the cursor.

To change a decision, type in a new number. Use the backspace key to correct numbers. Hit the enter key to check if the number is interpreted the way you had intended. The enter key is not necessary for game play.

When you have made your decisions hit

F4 Go to advance to the next day of the game.

Pause. Hit F2 when ready.
THE DISTRIBUTION GAME (MODIFIED)

DAY: 0

<table>
<thead>
<tr>
<th>Sales</th>
<th>$ 0.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.I.S.</td>
<td>0.00</td>
</tr>
<tr>
<td>Order Cost</td>
<td>0.00</td>
</tr>
<tr>
<td>Holding Cost</td>
<td>0.00</td>
</tr>
<tr>
<td>Net Profit</td>
<td>$ 0.00</td>
</tr>
</tbody>
</table>

Select screen type and game version.
[F4 Recall]  [F6 Color]  [F8 Game 2]  [F1 Help]  [F2 Return]

Reset Game

The game is automatically reset to its starting conditions when you leave this menu and return to the Main menu.

The following keys are active on this menu.

F4 Recall: a set of parameters from the disk. Your instructor may have created a special set of parameters for game competition.

F5 B&W: Select the Black and White version of the game.

F6 Color: Select the Color version of the game.

F7 Game 1: Select Game version 1 (full information).

F8 Game 2: Select Game version 2 (suppress display of retailer inventory levels until they fall below order point).

Pause. Hit F2 when ready.

F2 Go on
In game version 1 we have provided a helper to suggest decisions. This is intended for use by players who have no one to compete against. The helper is available only after you have entered the password and gained access to the Edit Parameters menu.

If the helper is available then you may hit:

F3 Helper to see what decisions the helper would make under the current circumstances.

F6 Auto to let the helper play the rest of the game by itself.

You could use F6 to get a score (net profit) against which you can compare your performance in the game.

---

Pause. Hit F2 when ready.  

---

For advanced players:

You can record your keystrokes and then replay them.

Hit "<ctrl> a", without the quotes, to turn ON recording.

Select function key under which to store the keystrokes.

Type menu name for keystrokes and hit carriage return or enter key. "Recording initiated"

From this point on your keystrokes are being recorded.

Hit "<ctrl> a", without the quotes, to turn OFF recording. "Recording completed"

Whenever you are ready, hit "<ctrl> z" to replay your keystrokes.

Pause. Hit F2 when ready.
Statistics

An "inventory day" is one unit of stock sitting in inventory or moving in transit for one day. Holding costs can be computed as

\[
\text{the number of inventory days} \times \text{purchase price per unit} \times \text{the annual holding cost rate} / \text{number of days in year.}
\]

A "backorder day" is one unit of stock being backordered for one day. For example, ten backorder days could mean one unit was backordered for ten days or ten units were backordered for one day.

Pause. Hit F2 when ready. |      |      |      |      |      |      | F2 Go on

| Total number of units sold | 1201 |
| Total number of units demanded | 1205 |
| Total number of warehouse orders | 5   |
| Total number of retailer shipments | 42  |
| Total number of warehouse inventory days | 25063 |
| Total number of retailer inventory days | 17646 |
| Total number of backorder days | 0   |

Do you have a CBA (graphics monitor)?

| F4 Yes |      |      |      |      |      | F1 Help | F2 No

A-14
### THE DISTRIBUTION GAME (MODIFIED)

**DAY: 200**

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales</td>
<td>$120100.00</td>
</tr>
<tr>
<td>C.G.S.</td>
<td>$84070.00</td>
</tr>
<tr>
<td>Order Cost</td>
<td>$1115.50</td>
</tr>
<tr>
<td>Holding Cost</td>
<td>$3386.16</td>
</tr>
<tr>
<td><strong>Net Profit</strong></td>
<td><strong>$31528.34</strong></td>
</tr>
</tbody>
</table>

**Edit shipments. Use tab key to select.**

- [ ] 16
- [ ] 3
- [ ] 8
- [ ] 383
- [ ] 397
- [ ] 425

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of units sold</td>
<td>1201</td>
</tr>
<tr>
<td>Total number of units demanded</td>
<td>1205</td>
</tr>
<tr>
<td>Total number of warehouse orders</td>
<td>5</td>
</tr>
<tr>
<td>Total number of retailer shipments</td>
<td>42</td>
</tr>
<tr>
<td>Total number of warehouse inventory days</td>
<td>25063</td>
</tr>
<tr>
<td>Total number of retailer inventory days</td>
<td>17646</td>
</tr>
<tr>
<td>Total number of backorder days</td>
<td>0</td>
</tr>
</tbody>
</table>
The Cumulative Plot

If you have a Color Graphics Adapter Card (CGA) in your PC then it is possible to have a graphical view of a time series that documents the impact of all the decisions you made in the course of playing the game.

Hit F4 if you have a CGA card and appropriate monitor. Otherwise, hit F2 to return to the Statistics menu.

The horizontal axis measures time, from day 0 to the ending day of the game. The vertical axis measures cumulative system inventory.

Three variables are plotted over time: (1) Cumulative Orders: this is the uppermost step function. It is initialized at day 0 to equal total system inventory. (2) Cumulative Shipments: this is the middle step function. It is initialized at day 0 to equal total retailer inventory. (3) Cumulative Sales: this is the lowermost step function. It is initialized at day 0 to equal zero.

In addition, material in transit is indicated by the shaded regions (the vertical stripes).

Pause. Hit F2 when ready. | | | | | | F2 Go on

The number of steps in step function (1) is the number of orders placed. The number of steps in step function (2) is equal to the number of shipments to retailers unless several shipments occur on the same day.

The vertical distance between step function (1) and step function (2) on any particular day is the amount of warehouse inventory, including orders in transit, on that day. The vertical distance between step function (2) and step function (3) on any particular day is the total amount of retailer inventory, including shipments in transit, on that day.

The area between step function (1) and step function (2) is equal to the total warehouse inventory days. The area between step function (2) and step function (3) is equal to the total retailer inventory days.

The average slope of step function (3) is the sales rate.

Pause. Hit F2 when ready. | | | | | | F2 Go on

A-15
Edit Game Parameters

The current game parameter is highlighted with the cursor.

To select a different game parameter, use the cursor control keys 
(<up>, <down>) to move the cursor up or down.

To change the current game parameter, hit

F3 to decrease the current value, or
F4 to increase the current value.

Each parameter has a minimum possible value, a maximum possible value, 
and a step size value that determines by how much you can increase it 
or decrease it. For example, the supplier to warehouse transit time can 
be no smaller than 15 days, no bigger than 45 days, and can only be 
changed in step sizes of 15 days.

Pause. Hit F2 when ready. |   |   |   |   |   | F2 Go on

Other keys:

F6 to change the password.

F7 to save the current set of parameters, including the password, 
to a disk file called "DIST.001". If a disk file with that name 
already exists, then you will be asked if you want to overwrite 
that file (destroying a previously saved set of parameters).
Type n if you do not want to overwrite and the save operation 
will be cancelled. Otherwise, type y.

To recall a previously saved set of parameters, see the Reset 
menu.

Note to instructors: set the initial random number to a different value, 
change the password, then save the parameters to a disk file. When your 
students begin the program, tell them to Recall parameters from the Reset 
menu and then begin to play the game. These steps will inhibit misuse 
of the game in competitions, but they cannot prevent it.

Pause. Hit F2 when ready. |   |   |   |   |   | F2 Go on

A-9
<table>
<thead>
<tr>
<th>R</th>
<th>Prob[DV+L&gt;R+1/2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.999552</td>
</tr>
<tr>
<td>1</td>
<td>0.998208</td>
</tr>
<tr>
<td>2</td>
<td>0.994624</td>
</tr>
<tr>
<td>3</td>
<td>0.986560</td>
</tr>
<tr>
<td>4</td>
<td>0.970816</td>
</tr>
<tr>
<td>5</td>
<td>0.943552</td>
</tr>
<tr>
<td>6</td>
<td>0.900928</td>
</tr>
<tr>
<td>7</td>
<td>0.840064</td>
</tr>
<tr>
<td>8</td>
<td>0.760320</td>
</tr>
<tr>
<td>9</td>
<td>0.663936</td>
</tr>
<tr>
<td>10</td>
<td>0.556032</td>
</tr>
<tr>
<td>11</td>
<td>0.443968</td>
</tr>
<tr>
<td>12</td>
<td>0.336064</td>
</tr>
<tr>
<td>13</td>
<td>0.239680</td>
</tr>
<tr>
<td>14</td>
<td>0.159936</td>
</tr>
<tr>
<td>15</td>
<td>0.099072</td>
</tr>
<tr>
<td>16</td>
<td>0.056448</td>
</tr>
<tr>
<td>17</td>
<td>0.029184</td>
</tr>
<tr>
<td>18</td>
<td>0.013440</td>
</tr>
<tr>
<td>19</td>
<td>0.005376</td>
</tr>
<tr>
<td>20</td>
<td>0.001792</td>
</tr>
<tr>
<td>21</td>
<td>0.000448</td>
</tr>
<tr>
<td>22</td>
<td>0.000064</td>
</tr>
<tr>
<td>23</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

Table 1. Approximate Stockout Probabilities for Various Order Points

**Fair Share Allocation**

The Game is structured to emphasize the opportunity the manager of a distribution system possesses in controlling the distribution of stock in the system. Rather than passively responding to orders placed by retailers, the player must actively decide where to place stock to maximize profit. Given a limited amount of warehouse stock, the player will frequently face the problem of how to allocate this stock, or a portion of it, among the different retailers. It should be immediately clear to the player that since the retailers are identical in all their basic characteristics (demand pattern, lead time, and cost and revenue structure) stock should be allocated to the retailers so that the resulting stock levels are as balanced as possible. In the literature of this problem, such an allocation is referred to as a fair share allocation. In models in which the retailers differ