Developments in Business Simulation and Experiential Learning, Volume 31, 2004

UNSORTING ALGORITHMS FOR AN ORDERED LIST AND ITS APPLICATION TO BUSINESS SIMULATIONS

Precha Thavikulwat
Towson University
pthavikulwat@towson.edu

Sharma Pillutla
Towson University
spillutla@towson.edu

ABSTRACT

When unbranded products are sold periodically in lots at auction to the highest bidder, the order in which the lots are placed into auction matters, because earlier lots generally will fetch higher prices than later lots. When the lots are ordered by their lowest reserve prices, suppliers who latch on to the equilibrium reserve price first should have a first-mover advantage that decays over time, as arranged by an unsorting algorithm. Such algorithms have three attributes: choice of target, distance of movement, and directional probability. The simple top-down, single-step, no-upward-movement algorithm is flawed. A compound algorithm may be satisfactory. Other algorithms may be more efficient. These ideas may have application in electronic business.

INTRODUCTION

Consider the situation wherein a number of firms of a computerized business simulation submit their terms of trade to a transaction-based market (Teach, 1990) on a periodic basis. In such a market, the products of firms are sold in lots. Thus, the products of a firm that submits a large quantity for sale may be sold in several small lots, each at possibly a different price. Likewise, the products of a firm that submits a small quantity for sale may be consolidated with those of its competitors, and sold in one large lot at a single price. Markets of this kind can be found in the agricultural and fishing industries, where the products that are brought to markets are not branded.

Unbranded products can be efficiently sold by auction to the highest bidder (Pillutla, 2002). In an auction, the order in which the lots are placed into auction is important, because the first lot meets with the greatest number of potential buyers, whereas the last lot meets with the fewest, because successful bidders of earlier lots will have already left the market. In the case when the products are sold through sealed bids that must be submitted before trading begins, the advantage to the lots entered into auction first is entirely compelling.

The problem that attends this type of market centers on the rule that determines the order in which the lots are to be put into the auction. Consider three rules: first-in-first-out (FIFO), last-in-first-out (LIFO), and lowest reserve price (LRP). In the analysis presented below, each of these rules will be considered with respect to random ordering, which may be though of as the no-rule rule.

FIFO, LIFO, and LRP

With FIFO, lots are placed into auction in the order of their arrival at market. In the everyday-world context, this can be undesirable because it encourages supplies to quit work sooner so that they may get to market earlier. As a result, less may be available to the market for sale. In the business gaming context, it encourages participants to spend less time on analysis, so that they may have their decisions done and submitted ahead of their competitors.

With LIFO, lots are placed into auction in reverse order of their arrival at market. As each competitor will want to be last before the deadline, LIFO promotes last-minute chaos and conflicts. Yet, as Thavikulwat (2003) has noted, LIFO keeps late-arriving suppliers from defecting to another market, where they may be at a lesser disadvantage. This phenomenon can be observed in the supermarket, where those who find themselves at the end of a long line at the checkout rush to get to the head of a new line when it opens.

With LRP, suppliers set a reserve price below which their products will not be sold in the auction. The lots are then ordered from lowest reserve price to highest reserve price. This method avoids the problems of FIFO and LIFO. It encourages firms to set low reserve prices because the lot with the lowest reserve price will actually fetch the highest bid price, because the lowest-reserve-price lot is sold first to the highest bidder. LRP, however, does not resolve the problem completely, for the same reserve price may be set by more than one firm. LRP must be supplemented by a secondary rule, possibly FIFO.

SUCCESSIVE PERIODS

Another problem to be resolved is how to order the lots if the same terms are submitted by the same firms in successive periods. If the preceding period's ordering is grand fathered, so that, in the case of LRP-FIFO, a firm's lot retains its place of the previous period, then the first-mover advantage of the firm that first engages the equilibrium reserve price to which all other competitive firms gravitate will be preserved forever, or at least until the end of the competition, unless reserved prices change. Late movers, having no hope of getting ahead, will be ill served by the market. In the everyday-world context, they will have reason to move their products to another market, which will cause prices in the original market to rise because of the lower supply, which in turn will cause buyers to place their bids in the other market, where prices would be lower. Accordingly, the grand fathering of LRP-FIFO cannot be sustained by the everyday-world market. In the gaming simulation context, grand fathering would accentuate the problem of early dominance, wherein the relative standing of companies changes little after the first few periods (Patz, 1992, 1999, 2000; Peach and Platt, 2000; Rollier, 1992).

An alternative to grand fathering is shuffling, that is, the order of the lots can be disturbed so that it is not perfectly replicated in successive periods. The objective is to give firms that engage the equilibrium reserve price first an advantage that decays gradually over time. This rewards the more capable firms and hastens the market's movement towards its equilibrium, thus enhancing its efficiency. Yet, it also gives late movers reason for staying with the market, because their disadvantage diminishes the longer they stay.

UNSORTING ALGORITHMS

Shuffling the order of the lots supplied to an auction is conceptually the same as unsorting an ordered list. If the order of the lots is to decay over a number of periods, the unsorting must be a process that only partially randomizes the items of the list. The unsorting algorithm can be conceived as having three primary attributes: choice of target, distance of movement, and directional probability.

Choice of target refers to the way an item is chosen for possible movement. The choice can be order based or item based. An order-based algorithm moves sequentially either top down or bottom up through the list, choosing whichever item happens be in the selected position of the list. An item-based algorithm chooses each item once and only once in a single pass. Thus, given the A-B-C ordered list, an order-based, top-down algorithm moves the first item first, then the second, and then the third. If the first movement results in B-A-C, because A moves down one step, the second movement will result in B-C-A if A moves down another step. Item-based, the second movement targets B, which moving down one step will result in A-B-C, thereby returning the list to its original order.

Distance of movement refers to the number of places a targeted item moves when it moves, the minimum being one place and the maximum, the length of the sorted list. Thus, in an alphabetically ordered list of three items, A-B-C, if A is targeted to move down two steps, it displaces C, which then takes A's place at the head of the list, giving rise to C-B-A. On the other hand, if A is targeted to move down one step, it displaces B, giving rise to B-A-C.

Directional probability refers to the likelihood a targeted item of the list will move in any one of three possible directions: up, in place, and down. *Up* means the item will move up the list, from perhaps second place to first place. *In place* means the item will retain its place on the list. *Down* means the item will move down the list, from perhaps second place to third place. Inasmuch as the three directions are exhaustive and mutually exclusive, the sum of the directional probabilities must be 1.0. Thus, if the targeted item has a 50-50 chance of moving up or down, but no chance of staying in place, the probabilities would be 0.5 up, 0.0 in place, and 0.5 down. If the targeted item has no chance of moving up, and a 50-50 chance of moving down or staying in place, then the probabilities would be 0.0 up, 0.5 in place, and 0.5 down.

DEGREE OF RANDOMIZATION

To find the set of attributes most suitable for any application, a measure of the degree of randomization that the algorithm accomplishes is needed. Given an ordered list of fixed length, the position of an item within the list averaged over a large number of runs of the algorithm is one such measure. If the algorithm completely randomizes the list, then the average position of every item will be at the midpoint of the list. The extent to which the lot has decayed from its initially sorted state after any iteration, or period, is measured by the average positions of two items: the item that was initially first of the list and the item that was initially last of the list. The unsorting algorithm has a symmetrical effect to the extent both items move towards the midpoint at the same rate.

ORDER-BASED TOP-DOWN, SINGLE-STEP, NO-UPWARD-MOVEMENT ALGORITHM

The order-based top-down, single-step, no-upward-movement algorithm (TDSS1) is especially simple. Applied to a list of *n* items, it requires *n*-1 steps, as illustrated in Figure 1. In Step 1, Item A is targeted. Depending on chance, it may stay in place or be moved down one step, displacing the second item, B. In Step 2, the item in the second position is targeted. Depending on chance, it also may stay in place or be moved down.

Figure 1
Order-Based Top-Down, Single-Step, No-Upward-Movement Algorithm Applied to Three Items

Initial State	Step 1	Step 2
A B C	A B C	A B C
		A C B
	B A C	B A C
		B C A

To obtain average positions, this algorithm was run 100 times each period for 20 successive periods on an alphabetically ordered list of 26 items. Five directional probability settings were applied, from 0.9 in place and 0.1 down, to 0.1 in place, and 0.9 down, in increments of 0.2. That is, the probability of downward movement increased in steps of 0.2 from 0.1 to 0.9, while the corresponding probability of staying in place decreased in the same steps from 0.9 to 0.1. The results are graphed in Figures 2 and 3. Probability labels on the graph refer to probabilities of downward movement.

Figure 2
Average Position of the First Item After 100 Simulated Runs

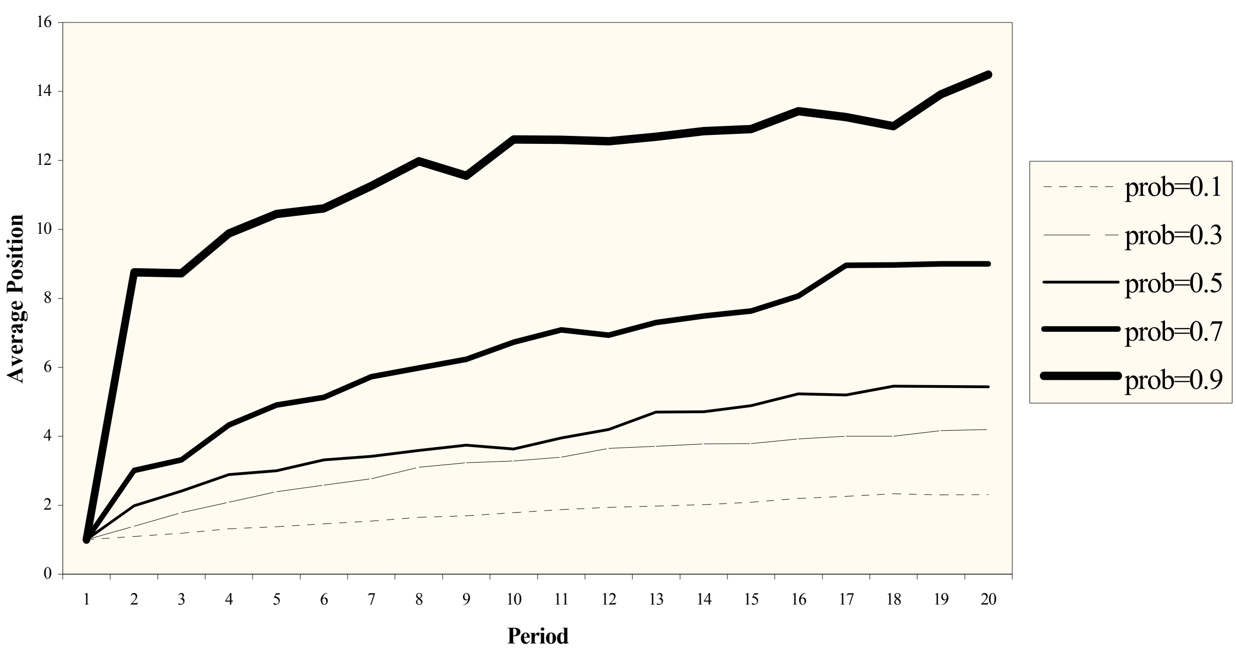
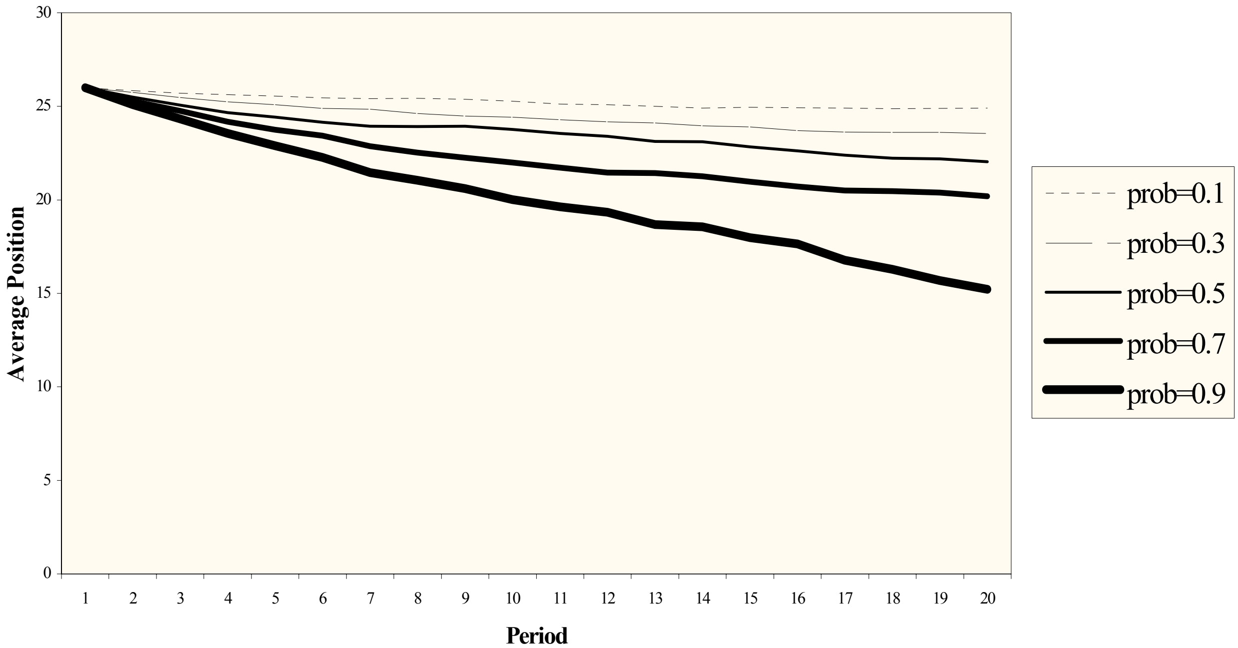


Figure 3
Average Position of the Last Item After 100 Simulated Runs



The graph of average positions of the first item is concave (Figure 2), whereas the graph of the average positions of the last is more linear (Figure 3), so TDSS1 has an asymmetrical effect. The contrast is especially striking for the downward movement probability of 0.9.

Asymmetry results because the top-down process can move the first item from the first position to the last position in one period, but it can only move the last item up one position at the most in a period. This is seen clearly in Figure 1, where the first item, A, can drop to the last position after the final step of a single iteration, but the last item, C, cannot advance more than one position in one iteration. Accordingly, high downward probability levels of TDSS1 can limit first-mover advantage to a few periods, but it requires late mover to endure the handicap of their low positions for much longer.

CONCLUSION

Unsorting algorithms are useful for situations requiring an ordered list that decays gradually to randomness over a number of periods. Such a list is essential in business simulations wherein products are sold through auction, lot by lot, to the highest bidders. The attributes of such algorithms have been delineated, and the simple order-based top-down, single-step, no-downward-movement algorithm has been shown to by asymetrical.

For transaction-based simulations that run for the typical length of a dozen periods or fewer (Anderson & Lawton, 1992), a compound order-based algorithm, combining the top-down, single-step, no-upward-movement algorithm with a bottom-up, single-step, no-downward-movement algorithm, might be satisfactory. In this case, the sorted list would be adjusted in two passes. The first pass would be top down, with no upward movement; followed by the second pass, which would be bottom up, with no downward movement. The result should be symmetrical for both ends of the ordered list.

The characteristics of item-based, multiple-step, and all-direction-movement algorithms have not been analyzed herein, but are certainty worthy of study. One of these may yield results equivalent to the compound algorithm, while being computationally more efficient. Moreover, the special requirements of a particular simulation may call for a specially constructed unsorting algorithm, which may or may not be defined by the three attributes that have been discussed herein.

Finally, lot-ordering rules and unsorting algorithms may have application to the emerging field of electronic business. If they find application there, they will be among the many ideas developed in artificial laboratory settings that have everyday-world applications.

REFERENCES

Anderson, P.H. & Lawton, L. (1992). A survey and methods used for evaluating student performance on business simulations. *Simulation & Gaming, 23*, 490-498.

Patz, A. L. (1992). Personality bias in total enterprise simulations. *Simulation & Games, 23*, 45-76.

Patz, A. L. (1999). Overall dominance in total enterprise simulation performance. *Developments in Business Simulation & Experiential Learning, 26*, 115-116.

Patz, A. L. (2000). One more time: Overall dominance in total enterprise simulation performance. *Developments in Business Simulation & Experiential Learning, 27*, 254-258.

Peach, E. B. & Platt, R. G. (2000). Total enterprise simulations and optimizing the decision set: Assessing student learning across decision periods. *Developments in Business Simulation & Experiential Learning, 27*, 242-247.

Pillutla, S. (2002). Investigation of the impact of decision parameters for a Dutch auction simulation for IPO issues. *Developments in Business Simulation & Experiential Learning, 29*, 149-154.

Rollier, B. (1992). Observations of a corporate facilitator. *Simulation & Gaming, 23*, 442-456.

Teach, R. D. (1990a). Designing business simulations. In J. W. Gentry (Ed.), *Guide to business gaming and experiential learning* (pp. 93-116). East Brunswick, NJ: Nichols/GP Publishing.

Thavikulwat, P. (2003). Gaming agency markets. Developments in Business Simulation Experiential Learning, 30, 225-230.