Intro

The mini tool "***GeneralKSearch***" is developed through an internship project in the field of online trading algorithms in the faculty of Information- and Technology Management at University of Saarland in winter semester 2011/2012.

The implemented online trading algorithm is based on the paper "Online algorithms for the general k-search problem" by Wenming Zhang et al.

The online time series search problem is introduced by El-Yaniv et al., in which a player is to sell one unit of some asset within several periods, aiming at the maximum selling revenue. At each period, the player observes a quoted price and decides immediately whether or not to accept the price, i.e., selling the unit of the asset at the price, without the knowledge of future prices. The game ends either one price is accepted or the last period is met.

Developer

Supervisor: Dr. E. Mohr

Sara Pourghourban

Problem Definition

The paper investigates the general k-search problem, in which a player is to sell totally k units of some asset within n periods, aiming at maximizing the total revenue. At each period, the player observes a quoted price which expires before the next period, and decides irrecoverably the amount of the asset to be sold at the price. A deterministic online algorithm is presented and proved optimal for the case where $k \leq n - 1$. For the other case where $k \geq n$, it is shown by numerical illustration that the gap between the upper and the lower bound of competitive ratio is quite small for many situations.

Here a general k-search problem is considered such that the player may sell more than one unit of the asset at each period and also are considered both cases where $k \leq n - 1$ and where $k \geq n$.

The general k-search problem generalizes the k-search problem by allowing selling multiple units of the asset at each period and eliminating the assumption of $k \leq n$. It is mainly presented an optimal deterministic algorithm (abbr. DET) for the case where $k < n$, and for the case where $k \geq n$, numerical computation shows that the gap between the upper and the lower bound is quite small for many situations.

So, a player is to sell $k \in N$ units of some asset in n periods, aiming to maximize the total revenue. At each period $i = 1, 2, \ldots, n$, the player observes a quoted price p_i between $[m, M]$ where n, m and M $(0 < m < M)$ are known beforehand. The player needs to decide irrecoverably in period i the number of the asset to be sold at price p_i without the knowledge of future prices, and all the k units of the asset shall be sold by period n.

Method

The DET algorithm:

The main idea of algorithm DET is to predetermine k reservation prices and then the number of the asset to be sold at each period depends on the relationship between the quoted price and a matching reservation price.

The k reservation prices $p_1^*, p_2^*, \ldots, p_k^*$ as well as $a(\geq 1)$ are the solution of the following $k + 1$ equations. For each formula on the left-hand side of each equation, the numerator denotes the upper bound of OPT's total revenue while the denominator denotes the lower bound of DET's for some quoted price input instance.

$(kp_1^*/km) = a,$

$(kp_2^*/(p_1^* + (k-1)m)) = a,$

$(kp_3^*/(p_1^* + p_2^* + (k-2)m)) = a,$

.

.

.

$(kp_k^*/(p_1^* + p_2^* + p_{k-1}^* + m)) = a,$

$(kM/(p_1^* + p_2^* + \ldots + p_k^*)) = a.$

By the first k equations, $p_i^* = [(1 + a/k)i - 1(a - 1) + 1]m$ for $(1 \leq i \leq k)$. Substituting the above solution of p*i into the last equation, we have with algebraic calculation that $(a - 1)(1 + a/k)^k - (M/m - 1) = 0$. Let $F(a) = (a - 1)(1 + a/k)^k - (M/m - 1)$. It can be verified that $F(1) < 0$ and $F(M/m) > 0$. We claim that there exists an a $(1 < a < M/m)$ such that $F(a) = 0$. Substituting the value of a into the solution of p_i^*, we determine the value of p_i^* which satisfies $m < p_1^* < p_2^* < \cdots < p_k^* < M$. For notational convenience, we virtually set $p_0^* = 0$ and $p_{k+1}^* = M$.

Algorithm DET:

At Period j (1 =< j =< n), a quoted price p j is observed. If $j = n$, sell all the remainder with price p_n. Otherwise assume that p j belongs to $[p_i^*, p_{i+1}^*)$ for some i where $(0 =< i =< k)$. DET behaves in two cases according to the values of $p_1, p_2, \ldots, p_{j-1}^*$ and p_i^*.

Case 1. $p_i^* > max\{p_1, p_2, \ldots, p_{j-1}^*\}$. Sell some units of the asset such that the accumulated sold amount is exactly i units.

Case 2. $p_i^* \leq max\{p_1, p_2, \ldots, p_{j-1}^*\}$. Sell none.

Let $m = 1, M = 10, k = 3$ and $n = 5$. We have by calculation that $a = 2.478, p_1^* = 2.478, p_2^* = 3.699, p_3^* = 5.929$. We virtually set $p_0^* = 0$ and $p_4^* = 10$. The quoted price instance is given in the second row of the table, and the last row demonstrates the selling process for DET. At Period 1, $p_1 = 3.45$ is in $[p \,\square\, 1, p \,\square\, 2)$ and thus DET sells one unit of the asset with price p_1. For Periods 2 and 3, DET sells none since Case 2 is met. At Period 4, since $p_3 \,\square\, [p_3^*, p_4^*)$and DET has already sold one before, it sells the rest two units with price p_3 at this period. The total revenue obtained by DET is equal to $p_1 + 2p_3 = 3.45 + 2 \times 7.72 = 18.89$.

Tool Operation

The tool operation begins with setting input parameters such as k, n, m , and M. Then we select the Excel file with the $.xls$ extension, for which we want to observe the Total Revenue. We just fill the first and second columns of sheet, in which each row represents the period and the price of that period, respectively. The labels of cells and the intermediate steps to calculate the total revenue would be exhibited on the input excel file and the output table.

Screenshots

Figure1. The Start window.

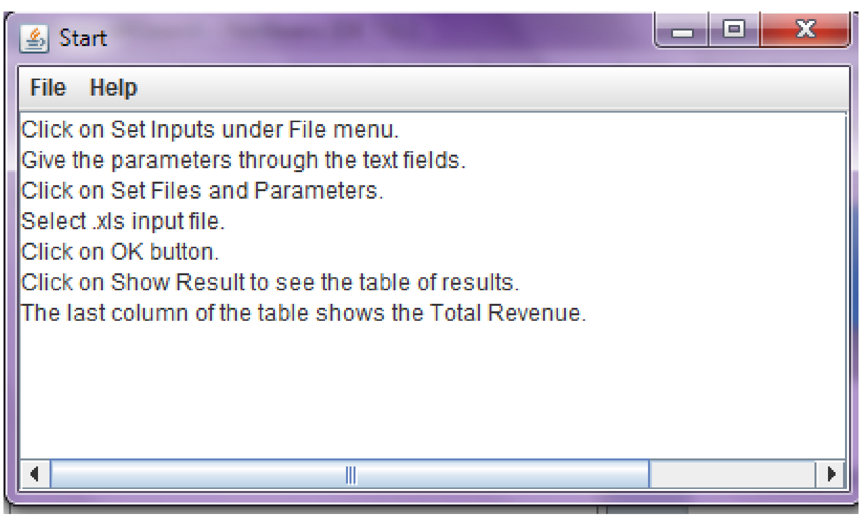


Figure 2. The select window for parameters and file.

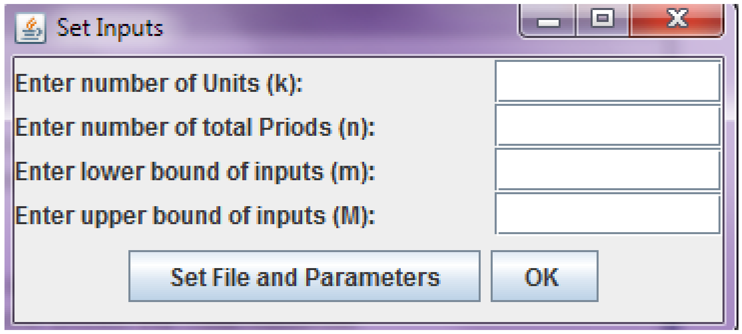
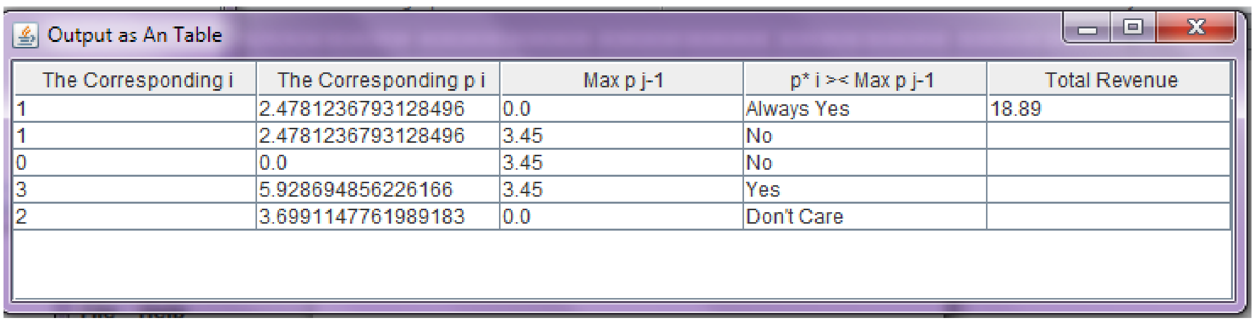


Figure 3. The Table of Results.



The Corresponding i	The Corresponding p i	Max p j-1	p* i >< Max p j-1	Total Revenue
1	2.4781236793128496	0.0	Always Yes	18.89
1	2.4781236793128496	3.45	No	
0	0.0	3.45	No	
3	5.928694856226166	3.45	Yes	
2	3.6991147761989183	0.0	Don't Care	

Terms of Use

//

Technical Requirements

- Java Runtime Environment (JRE) which is installed on the System to run the tool which is jar file.
- Excel 97-2003 to prepare the inputs for the tool.

Literature

W. Zhang, Y.Xu, F.Zheng, and M.Liu, *Online algorithms for the general k-search problem*, Elsevier B.V., 2011.

R. El-Yaniv, A. Fiat, R.M. Karp, G. Turpin, *Optimal search and one-way trading online algorithms*, Algorithmica 30 (2001) 101–139.