Information Processing Letters 113 (2013) 188–192

Contents lists available at SciVerse ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Online inventory replenishment scheduling of temporary orders

Haiyan Yu [a,b,*], Yinfeng Xu [a,b], Tengyu Wu [a,b]

[a] *School of Management, Xi'an Jiaotong University, Xi'an 710049, China*
[b] *The State Key Laboratory for Manufacturing Systems Engineering, Xi'an 710049, China*

ARTICLE INFO

Article history:
Received 14 October 2012
Received in revised form 22 December 2012
Accepted 31 December 2012
Available online 4 January 2013
Communicated by F.Y.L. Chin

Keywords:
Replenishment
Scheduling
Online algorithm
Competitive analysis

ABSTRACT

This paper studies the online inventory replenishment scheduling problem, in which orders from retailers arrive and depart at arbitrary times at one supplier and each order needs to be replenished at least once every constant periods. The supplier needs to assign each order to some vehicle such that the maximum number of vehicles ever used over all time is minimized, while each vehicle can replenish only one order per period. We present a Combined First Fit algorithm and improve the upper bound of the competitive ratio for this problem to be 3.0913 from 5. When offline algorithm is unrestricted, the Combined First Fit algorithm is proved to be 6.1826-competitive.

1. Introduction

We consider a two-stage supply chain system in which one supplier supplies an item to several retailers. Suppose retailer i has capacity q_i and a deterministic demand D_i per period. It is assumed that shortages are not allowed at each retailer. Thus retailer i should be replenished at least once per $r_i = \lfloor \frac{q_i}{D_i} \rfloor \geqslant 1$ periods. We call r_i as the *replenishment interval* of retailer i. Of course it is allowed to replenish retailer i once less than r_i periods. The supplier controls a fleet of vehicles with the same capacity q which is large enough for any replenishment, and each vehicle can replenish only one retailer per period. Order w_i from any retailer arrives and departs at arbitrary times, which corresponds to a triple (a_i, d_i, r_i) and a_i is the arrival time, d_i is the departure time. We assume that $r_i \geqslant 1$ for all i, and $a_1 \leqslant a_2 \leqslant \cdots \leqslant a_n$. The supplier needs to assign each order to some vehicle which is to be decided. Migration is not allowed, in the sense that once an order is assigned to a vehicle, it cannot be re-assigned to another vehicle. The objective is to minimize the maximum number of vehicles ever used over all time.

This research can be applied in the replenishment of some seasonal goods (such as fruit, ice-cream and so on) to convenience stores (like 7-Eleven), or the replenishment of barreled drinking water to houses. Another application can be found in push-based broadcasting systems, as it's mentioned in Chan and Wong's research [1].

There are two versions of this problem: online and offline inventory replenishment scheduling problems. In the online version, orders arrive over time, i.e., after the supplier has scheduled all the orders given so far, the information of the next order is known. Note that the supplier only knows r_i at the arrival a_i, but knows d_i immediately after the last replenishment for this order. In the offline version, the whole information about w_i, i.e., all the values of a_i, d_i, r_i are known before scheduling.

Given an input sequence $W = (w_1, w_2, \ldots, w_n)$ and an online algorithm A, let $A(W, t)$ denote the number of vehicles used at time t, and $A(W) = \max_{0 \leqslant t \leqslant a_n} A(W, t)$ is the maximum number of vehicles ever used over all time. Let $OPT(W)$ denote the maximum number of vehicles ever

* Corresponding author at: School of Management, Xi'an Jiaotong University, Xi'an 710049, China. Tel.: +86 13468976154; fax: +86 29 82665034.
E-mail addresses: yuhai8543@stu.xjtu.edu.cn (H. Yu), yfxu@mail.xjtu.edu.cn (Y. Xu), fly200205@163.com (T. Wu).

0020-0190/$ – see front matter
http://dx.doi.org/10.1016/j.ipl.2012.12.011

Table 1
Comparison of the results.

		Chan et al.'s [1,9]	Ours
Upper bound	$\min_i r_i \geqslant 1$	5	3.0913
	$\min_i r_i \geqslant 2$	4	2.6
	$\min_i r_i \geqslant 3$	4	2.4
	$\min_i r_i \geqslant k$, $k \geqslant 4$	$2 + 2/(\alpha^* - 1)$	$4k/(3k-4)$
Lower bound		2.428	

α^* is the largest power of 2 smaller than or equal to k.

used by optimal offline algorithm. If there exist two constants α and β such that $A(W) \leqslant \alpha \cdot OPT(W) + \beta$ stands for any input W, we call α as the competitive ratio and algorithm A is α-competitive [2]. We use the competitive ratio α to measure the performance of online algorithm A. The smaller α is, the better online algorithm A performs.

This study is motivated by the study of Li et al. [3] in which the inventory replenishment scheduling problem to minimize the number of vehicles was proposed and the conditions of existing feasible scheduling of one vehicle were studied. It's a complement to the study of inventory routing problem (IRP) [4], and the efficiency of IRP with direct delivery, in which one vehicle can replenish only one retailer per period, was studied [5–7]. Some researchers [1,8] studied the similar model, which was applied in the field of satellite communication and denoted as windows scheduling. Besides, this problem is distinguished from the dynamic bin packing problem of unit fractions items [9] by the difficulty of finding fit bin (or vehicle) for a given item (or order).

Previous results. When the offline algorithm is restricted to maintain the *load* of every vehicle ($\sum_{r_i} 1/r_i$) to be at most 1 at any time, Chan and Wong [1] gave a 5-competitive online algorithm and showed that there was a lower bound of $2 - \varepsilon$ for any $\varepsilon > 0$ on the competitive ratio of any online algorithm. Then Chan et al. [9] improved the lower bound to be 2.428 in the study of the dynamic bin packing problem with unit fraction items. The bounds are shown in Table 1.

For given $S = \{r_1, r_2, \ldots, r_n\}$, $r_1 \leqslant r_2 \leqslant \cdots \leqslant r_n$, Li et al. [3] proved that if there exists $\sum_{j=1}^{k} \frac{n_j}{2^{j-1} r_1} \leqslant 1$, S is *schedulable* by one vehicle, i.e. we can use one vehicle to replenish these n retailers, where n_j is the number of the elements whose values lay in $[2^{j-1} r_1, 2^j r_1)$ and $k = \lceil \log_2 r_n / r_1 \rceil$. Fishburn and Lagarias [10] studied a similar model as [3], and proved S is schedulable if the *load* of one vehicle $\sum_{i=1}^{n} 1/r_i \leqslant 3/4$ when $r_1 \geqslant 3$ and if the *load* of one vehicle $\sum_{i=1}^{n} 1/r_i \leqslant 5/6$ when $r_1 = 2$.

Our contribution. When the offline algorithm is restricted to maintain the load of every vehicle to be at most 1 at any time, we present a Combined First Fit algorithm and prove its competitive ratio to be 3.0913 for the case $\min_i r_i \geqslant 1$, which improves the upper bound of this problem greatly. For the cases $\min_i r_i \geqslant k$, $k = 2, 3, \ldots$, the upper bounds can be improved correspondingly (see Table 1). When offline algorithm is unrestricted, we analyze the property of the maximum value of one vehicle's load and the Combined First Fit algorithm is 6.1826-competitive.

2. Combined First Fit algorithm

In this section, we present the Combined First Fit algorithm to solve the online inventory replenishment scheduling problem and get a new upper bound by competitive analysis.

Combined First Fit algorithm (CFF). Assume orders $w_1, w_2, \ldots, w_{i-1}$ have been scheduled, and the occupied vehicles are $v_1, v_2, \ldots, v_{j^*}$. Let $S_j = \{r_{j1}, r_{j2}, \ldots, r_{jn_j}\}$, $r_{j1} \leqslant r_{j2} \leqslant \cdots \leqslant r_{jn_j}$ denote the set of replenishment intervals of the orders which are replenished by vehicle v_j at time t for any $1 \leqslant j \leqslant j^*$. Note that $a_{ji} \leqslant t < d_{ji}$. For a new arrival order w_i, we check whether the vehicles fit for the new order starting from the lowest index. If r_i and $S_j = \{r_{j1}, r_{j2}, \ldots, r_{jn_j}\}$ satisfy any one of the following three conditions, vehicle v_j is fit for order w_i. Try to put r_i into set S_j, refresh $S_j = \{r_{j1}, r_{j2}, \ldots, r_{jn_j}, r_{j(n_j+1)}\}$, $r_{j1} \leqslant r_{j2} \leqslant \cdots \leqslant r_{jn_j} \leqslant r_{j(n_j+1)}$.

(1) Let $S_{j1} = \{r_{j1}, \ldots, r_{jm_1}\}$, $r_{jm_1} < 2r_{j1} \leqslant r_{jm_1+1}$;

$$S_{j2} = \{r_{jm_1+1}, \ldots, r_{jm_2}\}, r_{jm_2} < 2^2 r_{j1} \leqslant r_{jm_2+1};$$

$$\ldots\ldots$$

$$S_{jk} = \{r_{jm_{k-1}+1}, \ldots, r_{jm_k}\}, r_{jm_k} < 2^k r_{j1} \leqslant r_{jm_k+1};$$

$$S_{j(k+1)} = \{r_{jm_k+1}, \ldots, r_{j(n_j+1)}\}, r_{j(n_j+1)} < 2^{k+1} r_{j1};$$

There exists $\frac{m_1}{r_{j1}} + \frac{m_2 - m_1}{2r_{j1}} + \cdots + \frac{(n_j+1) - m_k}{2^k r_{j1}} \leqslant 1$.

(2) There exist $r_{ji} = 2, r_{ji} \in S_j$ and $\sum_{i=1}^{n_j+1} \frac{1}{r_{ji}} \leqslant 5/6$.

(3) There exists $\sum_{i=1}^{n_j+1} \frac{1}{r_{ji}} \leqslant 3/4$.

If vehicle v_j is fit for order w_i, assign vehicle v_j to replenish for w_i. Otherwise, assign w_i to a new vehicle v_{j^*+1}. If vehicle v_j is fit for order w_i due to condition (1), schedule $S_j = \{r_{j1}, r_{j2}, \ldots, r_{jn_j}, r_{j(n_j+1)}\}$ by modifying the values of all the elements in S_{jh}, $1 \leqslant h \leqslant k+1$ into $2^{h-1} r_1$, and schedule the replenishment according to the modified replenishment intervals one by one [3]. If vehicle v_j is fit for order w_i due to condition (2) or (3), replenish order w_{j1} every r_{j1} periods, and schedule the replenishment of the rest orders one by one as following: replenish w_{jh}, $2 \leqslant h \leqslant n_j + 1$ every r'_{jh} unoccupied periods (i.e. the period at which vehicle v_j hasn't been assigned to replenish any order is called unoccupied period), where r'_{jh} is the minimum number of unoccupied periods every r_{jh} periods [10].

Note. If vehicle v_j is fit for order w_i, we assume there exists one scheduling for vehicle v_j such that any shortage won't occur in the process of scheduling transition, which may be caused by the arrival of new order w_i. Because three conditions above correspond to different scheduling algorithms, the scheduling for vehicle v_j may be changed from one to another due to the new order's arrival. We suppose that there exists one scheduling to smooth the transition process. But a strict proof is needed for further research.