Robotics and Computer-Integrated Manufacturing 27 (2011) 62–71

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

# Online rescheduling of multiple picking agents for warehouse management

Jose I.U. Rubrico [a], Toshimitu Higashi [b], Hirofumi Tamura [c], Jun Ota [d,*]

[a] Department of Precision Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
[b] Murata Machinery, Ltd., 2 Nakajima, Hashizume, Inuyama-shi, Aichi 484-0076, Japan
[c] Murata Systems, Ltd., 3 Minamiochiai-cho, Kisshoin, Minami-ku, Kyoto 601-8326, Japan
[d] Research into Artifacts, Center for Engineering (RACE), The University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa, Chiba 270-8568, Japan

ARTICLE INFO

*Article history:*
Received 18 January 2008
Received in revised form
3 June 2010
Accepted 3 June 2010

*Keywords:*
Dynamic scheduling
Warehouse management
Online algorithm

ABSTRACT

In this paper, we present a solution for a dynamic rescheduling problem involving new orders arriving randomly while static orders have been given in advance in warehouse environments. We propose two variations of an incremental static scheduling scheme: one based on the steepest descent insertion, called OR1, and the other, on multistage rescheduling, called OR2. Both techniques are enhanced by a local search procedure specifically designed for the problem at hand. We also implemented several existing online algorithms to our problem for evaluative purposes. Extensive statistical experiments based on real picking data indicate that the proposed methodologies are competitive with existing online schedulers and show that load-balancing algorithms, such as OR1, yield the best results on the average and that OR2 is effective in reducing the picking time when dynamism is low to moderate.


## 1. Introduction

The problem of picking customer orders from warehouse storage to the packing/delivery area, i.e., order picking, is one of the most important activities in a warehouse due to its high operational cost. Studies have shown that order picking can amount to 60% of the total costs of warehouse operations [1]. Traditionally, due to the high percentage of travel time in an order picker's activity [2], order picking has been treated as a distance-minimization problem. Optimization of a single picking route may be found in Refs. [3–7], among others.

For multiple routes and order pickers, exact methods are presented in Refs. [8,9], while heuristics for the cluster-then-route approach are investigated in Ref. [10]. In a series of works, Rubrico et al. [11–14] solve a more complicated form of the order picking problem with multiple pickers (agents), in which each agent may be assigned more than one trips (routes) to accomplish and delays due to congestion or queuing are considered. The objectives are to minimize an agent's idle time by minimizing travel distance and delays as well as maintain the balance of load among the agents. They adapt a hierarchical decomposition and solve each subdivision using heuristics to achieve good schedules quickly for realistically sized problems.

The works reported above address a static order picking problem in which all orders to be picked are known beforehand. Purely online versions of a picking problem in which all orders are not known beforehand and arrive at random are addressed in Refs. [15,16], where stochastic distributions are assumed for the order parameters and the problem is solved using methods related to the queuing theory. The online picking problem can be seen as an instance of the general online scheduling problem, in which the agents retrieving the orders correspond to the machines that process the jobs. A description of the online scheduling problem and its variants, as well as surveys of eminent online algorithms, may be found in Refs. [17–20]. Online scheduling on uniform machines (i.e., a given job may be processed on any machine at the same cost) has been relatively well studied [21–25]. However, such a straightforward representation cannot be directly applied to the online picking problem because the cost incurred by the agent as a result of a random order will depend on the agent's current load and is further complicated by the occurrence of delays. A more appropriate representation of the online picking problem is the scheduling of unrelated machines, in which the cost of processing a given job may differ arbitrarily from one machine to another. Literature on this difficult variation of the online scheduling problem remains relatively sparse [26,27]. In one study, Aspnes et al. [26] investigate the characteristics of a post-greedy variation of Graham's algorithm (originally intended for scheduling on uniform machines [21]) and propose a procedure that attains the best possible worst-case behavior (according to competitive analysis [28,29]) for the online scheduling of jobs on unrelated machines.

Another problem closely related to the online picking problem is the dynamic vehicle routing problem (DVRP). Typical dynamic events include: (a) customer additions/deletions, (b) changing-path costs (e.g., road traffic status), and (c) contingencies, such as

* Corresponding author. Tel.: +81 4 7136 4252; fax: +81 4 7136 4242.
*E-mail address:* ota@race.u-tokyo.ac.jp (J. Ota).

0736-5845/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.rcim.2010.06.011

vehicle breakdowns and recovery. These dynamic parameters can cause re-planning of routes and rescheduling of vehicles as well as the addition of vehicles for dispatching to meet strict due dates. Their correspondence with the online picking problem is apparent, and, in it, the addition of customers corresponds to the random arrival of new orders. A survey of the classifications, variants, and methodologies for the DVRP may be found in Ref. [30]. In Ref. [31], a parallel tabu search is used incrementally to improve the solution between dynamic events (e.g., order of arrivals). A generalization of this approach is discussed in Ref. [32] as the multiple plan approach (MPA), which keeps multiple plans each time an event occurs to allow for greater flexibility in the solution choice as time progresses. MPA is a framework that is independent of the search procedure. In the same paper, the authors also introduce an enhancement of the MPA that includes the forecasting capability. This approach is referred to as the multiple scenario approach, in which the solutions kept include predicted demands incorporated with existing ones. In Ref. [33], a reactive architecture based on constraint programming is proposed to deal with dynamic changes in real time, while, in Ref. [34], a software prototype is presented that demonstrates the capabilities of reactive vehicle routing.

In this paper, we are interested in solving the dynamic problem of new orders arriving randomly while the static orders are in the process of being picked. This problem is referred to here as the online rescheduling problem because it involves modifying the existing agent schedules in order to effectively incorporate the new orders along with the old ones, which are yet unpicked. The arrival of new orders is considered in this study because it is the primary cause of dynamicity in a real picking operation. The rescheduling problem differs from strict online scheduling, in which all orders arrive randomly. The methods for online scheduling are somewhat limited for the rescheduling problem because they are unable to modify the existing schedule by redistributing existing trips among the agents. On the other hand, the DVRP with stochastic demands is more similar to the problem presented in this paper. However, because of the metropolitan scales of distances in the DVRP, a calculation time of approximately 1 min is acceptable from the methodologies reported in the preceding paragraphs. Such times are too slow for the picking problem and will severely delay operations. Thus, there is a need for faster rescheduling methods for our problem which, nonetheless, yield good and robust schedules.

The objective of the study is to investigate the performance of two variants of an incremental static scheduler that resolves the total schedule every time a new order arrives by comparing it to existing online schedulers in the literature and in general practice. The incremental static framework is adapted because of its speed and low resource overhead as compared with more elaborate methods, such as those in Refs. [31,32].

The rest of the paper proceeds as follows. In Section 2, the online rescheduling problem is described. In Section 3, the proposed and reference methodologies are described. In Section 4, results from evaluative experiments are presented, and Section 5 is the conclusion.

## 2. Problem

The online rescheduling problem is a dynamic extension of the warehouse picking problem (WPP) preliminarily described in Ref. [11]. In the static version, a set of customer orders must be retrieved from warehouse storage shelves by a group of picking agents and transferred to a common packing shed. In the online problem, new orders randomly arrive and are added to the list of orders to be picked even while the static orders are being retrieved. Fig. 1 illustrates an input (order list) and output (agent picking schedule) example. For the static problem, the order list remains unchanged in time until picking is completed, whereas, for the dynamic WPP (DWPP), the order list continually grows in a random manner as time passes until some terminating condition is satisfied. Some assumptions on the dynamic problem are enumerated below:

- static and dynamic orders have the same priority,
- all static and dynamic orders must be picked,
- a dynamic order cannot be inserted into a trip currently being traversed by an agent,
- only order additions are considered (i.e., no cancellations),
- a newly arrived order is composed of one type of product only.

### 2.1. Stochastic orders

The parameters that characterize the random orders are (a) arrival rate $\lambda$ (orders per unit time), (b) product information, i.e., storage quantity $p_{rand}$ and required quantity $d_{rand}$, and (c) total item ratio with static orders, $\gamma = D_{rand}/D_{static}$, where $D_{rand}$ is the total quantity of random orders picked and $D_{static}$ the total number of static orders.

The arrival of new orders is taken to be exponentially distributed. A transformation can be made to express the inter-arrival time in terms of the uniform distribution given in (1) below. The term $U$ denotes a uniform distribution of random variates within interval (0,1), while $\tau_{arr}$ is the inter-arrival time, i.e., the time interval between order arrivals:

$$\tau_{arr} = \frac{-\ln(U)}{\lambda} \tag{1}$$

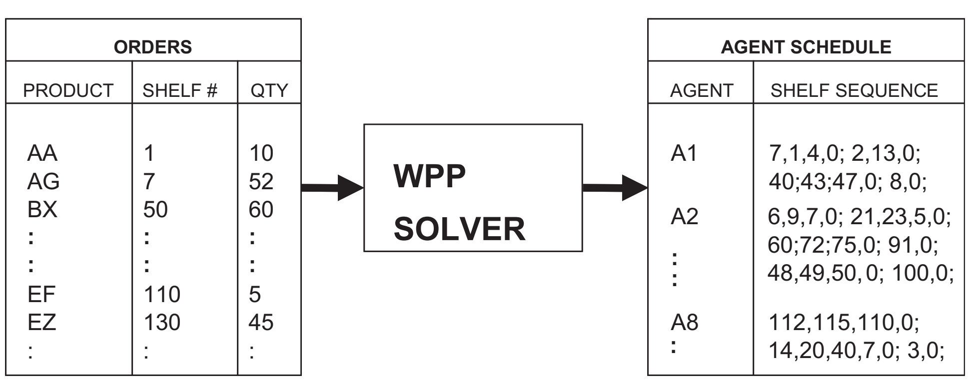


**Fig. 1.** Input and output of the picking problem.