European Journal of Operational Research 236 (2014) 968–977

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

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

A fast simulated annealing method for batching precedence-constrained customer orders in a warehouse

Marek Matusiak [a,*], René de Koster [b], Leo Kroon [c], Jari Saarinen [a]

[a] Finnish Centre of Excellence in Generic Intelligent Machines Research, Aalto University, P.O. Box 15500, 00076 Aalto, Finland
[b] Department of Management of Technology and Innovation, Rotterdam School of Management, Erasmus University, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
[c] Department of Decision and Information Sciences, Rotterdam School of Management, Erasmus University, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands

ARTICLE INFO

Article history:
Available online 13 June 2013

Keywords:
Heuristics
Logistics
Order picking
Batching
Precedence

ABSTRACT

Batching customer orders in a warehouse can result in considerable savings in order pickers' travel distances. Many picker-to-parts warehouses have precedence constraints in picking a customer order. In this paper a joint order-batching and picker routing method is introduced to solve this combined precedence-constrained routing and order-batching problem. It consists of two sub-algorithms: an optimal A^*-algorithm for the routing; and a simulated annealing algorithm for the batching which estimates the savings gained from batching more than two customer orders to avoid unnecessary routing. For batches of three customer orders, the introduced algorithm produces results with an error of less than 1.2% compared to the optimal solution. It also compares well to other heuristics from literature. A data set from a large Finnish order picking warehouse is rerouted and rebatched resulting in savings of over 5000 kilometres or 16% in travel distance in 3 months compared to the current method.

1. Introduction

Order picking is the single most important process in distribution centers. It is also the most laborious one, responsible for a substantial part of the distribution center's total costs, with estimates as high as 55% (Drury, 1988) to 65% (Coyle, Bardi, & Langley, 1996) mentioned in the literature. Although advances in technology have made the picking process more efficient and the actual percentage of the total costs is most probably less than the figures above indicate, it is still substantial. Travel time is the largest component of the order picking time, with a contribution of up to 50% (Tompkins, White, Bozer, & Tanchoco, 2003). Much research has been devoted to methods reducing order pickers' travel times, including modifying warehouse layout, using a different storage policy, batching orders, and routing pickers. In this paper, we focus on combined picker routing and order batching.

Our study is inspired by a picker-to-parts order picking process in a large retail warehouse in Finland. In this warehouse, at most three *customer orders* (or *orders* for short) can be batched in a single pick tour, each possibly for a different customer (i.e., a store) and a different delivery location within the warehouse. Each customer order has to be picked in a fixed sequence, due to family grouping in the company's stores. The problem of batching orders and routing pickers while respecting precedence constraints of products is common, particularly in retail organisations, but such restrictions may also play a role in other warehouses (Dekker, De Koster, Roodbergen, & Van Kalleveen, 2004; Chan & Kumar, 2008). Precedence constraints may vary in nature. They may be due to weight restrictions (heavy products at the bottom of the roll container), fragility (light at the top), shape and size (big boxes at the bottom), stackability, but also preferred unloading sequence due to family grouping on the customer's shelves. Although order batching and picker routing have received quite some attention in the literature, the combined problem of order batching and picker routing while respecting the precedence constraints of the products (and in our problem, including potential multiple drop-off points in a route) has not received much attention.

Batching orders and routing pickers while respecting precedence constraints is complex. Order batching is NP-hard for batches of size three or higher (Gademann & Van de Velde, 2005). Routing pickers while respecting precedence constraints has complexity of $O(N^2(P+1)^N)$ (Psaraftis, 1980a), where N is the number of customer orders in the batch and P is the maximum number of items in an order. Finding exact solutions for large orders and large batches becomes rapidly intractable. Our reference company has 2000 orders per day and up to 50 products per order, which have to be batched in batches of three orders. As the problem has to be solved multiple times per day and because of the problem's complexity, exact computation of the order batching is not feasible and thus a heuristic method is used.

* Corresponding author.
E-mail addresses: marek.matusiak@aalto.fi (M. Matusiak), rkoster@rsm.nl (R. de Koster), lkroon@rsm.nl (L. Kroon), jari.saarinen@aalto.fi (J. Saarinen).

0377-2217/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2013.06.001

We solve the problem using a generic solution method. For the batching, we use a simulated annealing-based combinatorial search algorithm based on maximising total savings in travel distance. Batching aims to reduce total travel distance by combining multiple similar orders (which have to visit overlapping parts of the warehouse). Selecting the orders to be batched is based on comparing all pairs of orders and estimating the savings due to combining a larger number of orders. This reduces computation times substantially, while the error compared to optimal combinations is within 1.2%. The routing can be solved using an A^*-type shortest path algorithm. The method is exact and general, not constrained by the particular warehouse layout.

Our contribution is a structured approach for the joint order batching and picker routing problem in warehouses with any layout or any strict sequence constraint. The method is fast, and compares well to optimal solutions and heuristics from literature. In our reference case, travel distance savings of nearly 16% could be achieved.

2. Literature review

A classic algorithm for order picking tour construction is the exact and polynomial time algorithm first presented in Ratliff and Rosenthal (1983) and extended in De Koster and Van der Poort (1998) to include multiple drop-offs. The algorithm presented in Ratliff and Rosenthal (1983) is further extended in Roodbergen and De Koster (2001b) to include a middle aisle. This extended version assumes a parallel-aisle warehouse with a maximum of three cross-aisles (see Fig. 1) and does not account for precedence constraints.

Next to optimal routes, heuristics are often used. In S-shape routing (Randolph, 1993), also known as traversal routing, an order picker travels the whole aisle if he or she needs to pick an item in the aisle visited. An exception is made for the last aisle if the number of aisles is odd. With largest-gap routing (Randolph, 1993), each aisle, except for the first and the last one visited, is exited on the entry side. The first and the last visited aisles are travelled completely. A heuristic combining S-shape and largest-gap routing is presented in Roodbergen and De Koster (2001a).

Theys, Braysy, Dullaert, and Raa (2010) consider the applicability of the Lin–Kernighan–Helsgaun (LKH) TSP heuristic (Helsgaun, 2000) to routing order pickers. They compare LKH with several routing heuristics and obtain savings of up to 47% in travel distance. Helsgaun (2000) finds optimal solutions with the LKH heuristic for all previously solved TSP instances available at that time, including a 13,509-city problem, which was the largest problem instance solved to optimality at the time.

A dynamic programming algorithm for solving the single-vehicle many-to-many immediate request dial-a-ride problem (multiple customer destinations, each with a possibly unique drop-off location), which is similar to our routing, is introduced in Psaraftis (1980b, 1983). The time complexity of Psaraftis' algorithm is

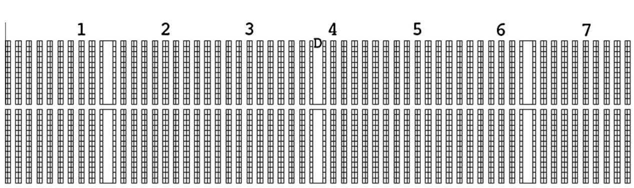

Fig. 1. Top view of the warehouse with seven drop-off locations indicated at the top (marked 1–7). The empty bin depot is denoted with D. The 57 aisles are unidirectional, while the three cross-aisles are bidirectional. The aisle-to-aisle distance is 5.5 meters the slot-to-slot distance is 3.7 meters, which is also the width of the central cross-aisle.

$O(N^2 3^N)$ (where n is the total number of customers), and it solves problems for up to ten customers.

Kubo and Kasugai (1991) introduce the Precedence-Constrained Travelling Salesman Problem (PCTSP) and a branch-and-bound method for finding exact solutions for cases of up to 49 locations with acceptable computation times. The precedence-constrained path construction problem can be modelled as a case of the Sequential Ordering Problem (SOP) (Escudero, 1988). The SOP can be formulated as an Asymmetric Travelling Salesman Problem (ATSP) with precedence constraints. In the SOP, paths usually have a start and a finish position that differ from each other, while ATSP paths finish where they started. In the general ATSP case, each of the non-visited cities can be the next target with each iteration, but in the SOP, the set of the next possible cities is limited as defined by a directed graph formed from the problem.

A method for batching orders is introduced in Gademann and Van de Velde (2005). The order-batching problem is modelled as a set partitioning problem. A column generation algorithm is used to solve the linear programming relaxation. They rely on the polynomial time algorithm presented in Ratliff and Rosenthal (1983) to calculate the route length. They find that the maximum batch size has the largest impact on the solution time.

The savings algorithm by Clarke and Wright (1964), C&W(i), as well as an extension of it, C&W(ii), are used for batching orders in De Koster, Van der Poort, and Wolters (1999). They are compared to *seed* algorithms using two routing strategies: S-shape and largest-gap. Seed algorithms consist of two distinct steps: seed order selection and order addition. A single order is selected as the seed order based on criteria, e.g., the highest number of items, longest travel time or the farthest item. Additions can be done using different rules such as adding the order that minimises the sum of the distances of every item of the seed and the closest item in the order, or minimising the additional number of aisles to be travelled. The authors find that seed algorithms work best with S-shape routing and large pick device capacity, while savings algorithms work best with Largest-gap and small pick capacity. C&W(ii) consistently outperforms C&W(i), but is computationally more expensive.

Hsieh and Huang (2011) introduce two batch construction heuristics based on data clustering method: K-means Batching (KMB) based on K-means algorithm (MacQueen, 1967); and Self-Organisation Map (SOMB) based on the Self-Organising Map (Kohonen, 1990). KMB functions in a manner similar to traditional seed algorithms, while SOMB uses the Self-Organising Map to choose batches for routing.

Albareda-Sambola, Alonso-Ayuso, Molina, and De Blas (2009) use a Variable Neighbourhood Search (VNS) algorithm to batch orders. It uses six different local exchange schemes incorporated into three different search neighbourhoods of varying size to find good batches. Within each neighbourhood, all moves belonging to that neighbourghood are tried, and the one which results in the largest savings is chosen. The larger neighbourhoods are searched as needed – if a current one fails to produce a better solution, the next (larger) neighbourhood is explored for a better solution until no improvement can be made. VNS is compared to the C&W(i), C&W(ii) and seed algorithms and it consistently outperforms them. The authors find that the best performing algorithm from literature is C&W(ii), which is on average 2% worse than VNS. Solution quality comes with added computational complexity – when compared to C&W(ii) a much larger part of the combinatorial batch space explored. For the most complex instance run (250 orders), VNS took almost six times as much time to reach its solution. For routing batches, the authors use the computationally inexpensive S-shape, largest-gap and combined heuristics.

Henn and Wäscher (2012) introduce an attribute based hill-climber (ABHC) (Whittley & Smith, 2004) for the order-batching problem. ABHC uses a set of attributes to guide the search out of