2018

Batching in Bucket Brigade Order Picking

Yossi Bukchin
Tel-Aviv University, bukchin@tau.ac.il

Eran Hanany
Tel Aviv University, hananye@post.tau.ac.il

Eugene Khmelnitsky
Tel-Aviv University, xmel@tau.ac.il

Batching in Bucket Brigade Order Picking

Yossi Bukchin
Dept. of Industrial Engineering
Tel-Aviv University
Tel-Aviv, Israel
bukchin@tau.ac.il

Eran Hanany
Dept. of Industrial Engineering
Tel-Aviv University
Tel-Aviv, Israel
hananye@post.tau.ac.il

Eugene Khmelnitsky
Dept. of Industrial Engineering
Tel-Aviv University
Tel-Aviv, Israel
xmel@tau.ac.il

***Abstract*— Order picking is the most labor cost consuming element in warehouse operations. In this paper, we consider an order picking process in a single picking aisle in the forward pick area, consisting of multiple locations (pick faces). The picking is performed by a group of pickers, each characterized by stochastic (forward and backward) walking process and picking times. We assume that the bucket brigade (BB) approach is applied, in a static environment, in which a given set of orders has to be picked. In order to improve the picking process, we suggest a batching procedure, where the objective is to minimize the total picking time, namely, the makespan. The proposed batching approach has two advantages: (1) it decreases the total travel time, since the items can be picked in a reduced number of picking tours as compared with picking each order separately; and (2) it balances the picking load along the picking aisle, consequently reducing the blockage occurrences. We model the problem as a Constraint Programming (CP) formulation, which was shown to be efficient in providing high quality solutions for non-linear models. Small and large scale examples are given to demonstrate the proposed approach, where the former consists of 24 orders, which are picked in five locations (pick faces), and the latter consists of 50 orders, which are picked in 12 locations. The solution obtained by the CP formulation is compared via simulation with an order by order picking and with a naïve batching approach, in which orders are batched in an arbitrary sequence until approaching the available capacity.**

Keywords—order-picking, bucket brigade, batching.

I. INTRODUCTION AND LITERATURE REVIEW

Order picking is the most labor cost consuming element in warehouse operations, as it consumes about 55% of the operating costs [1]. The travel time is the largest element of order picking, as it takes about 50% of its time. We can distinguish between order picking within the backward and the forward storage area. In the backward storage area, items are stored in large quantities, typically in large storage units (pallets or cartons) in multiple aisles. This implies on the order picking process, which may involve long travel using forklifts or electric/manual carts. The forward storage area (also called fast pick area), which is located close to the staging area, accommodates small quantities of items, located close to each other. In this area, orders are typically picked manually in piece units, while the travel (or walking) distances are relatively small. Most studies focuses on the backward storage area, as issues related to the order picking problems are batching orders, sequencing orders/batches, routing, and zoning [2]. Batching orders under given two routing strategies is investigated in [3]. The storage assignment and pick routing in backward pick area was studied in [4]. Gademann and Velde [5] proved that order batching problem in a parallel-aisle warehouse is NP-hard. An efficient heuristic for the order batching in parallel-aisle is then proposed in [6], which is compared with a lower bound obtained by an LP relaxation. Hong et al. [7] presents a MILP formulation along with simulated annealing for the batching-sequencing problem in parallel-aisle, and analyze the trade-off between wide and narrow aisle systems. When the single flow rack is concerned, typically in a forward pick area, a bucket brigade (BB) system, which was presented in [8], is commonly applied. This approach aims to reduce the blockage and starvation phenomena, as the latter is typically eliminated (depending on the model assumptions), and the former can be reduced by locating the workers/pickers in a slowest to fastest order. The index batching model for bucket brigade was suggested in [9], while using a Mixed Integer Programming (MIP) to minimize blocking delays among pickers. An analytical model for the blocking delay for two extreme walk speed cases is studied in [10], as they show that blocking increases with the workload imbalance along the line.

The current study addresses the batching problem in a single flow rack under bucket brigade protocol. The studied environment, depicted in Fig.1, addresses a single picking aisle consisting of multiple locations (pick faces). The picking is done by a group of pickers, each characterized by stochastic (forward and backward) walking process and picking times. Although the applied bucket brigade system aims to minimize blocking and starvation, still, the fluctuations in the workload along the picking aisle, together with the stochastic nature of the picker behavior may cause significant idle time. In this study, a static environment is considered, in which a given set of orders has to be picked. Each order is expressed by a vector of integers, where each element denotes the number of picks in the corresponding pick face. We suggest a new approach for batching these orders, where the objective is to minimize the total picking time, namely, the makespan. The proposed batching approach has two advantages: (1) it decreases the total travel time, since the items can be picked in a reduced number of picking tours (cycles in this case) as compared with picking each order separately; and (2) it balances the picking load along the picking aisle, consequently reducing the blockage occurrences.

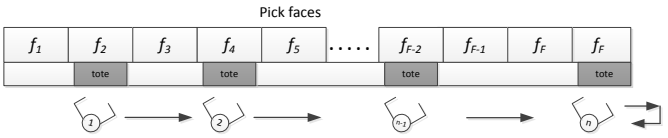

Fig.1. Problem environment

The proposed batching procedure aims to utilize the picker capacity and at the same time, to generate homogeneous batches. To this end, we develop and solve a Constraint Programming (CP) formulation. CP has some advantages over Mixed Integer Programming: (1) it can provide high quality solutions, for large scale instances, relatively fast, and (2) it can solve problems with non-linear objective/constraints. Other advantages of CP, such as using logical constraints, global constraints, decision variables as indices, are not needed here. Small and large scale examples are given to demonstrate the proposed approach, where the former consists of 24 orders, which are picked is five pick faces, and the latter consists of 50 orders, which are picked in 12 pick faces. The solution obtained by the CP formulation is first compared via simulation with an order by order picking. Then it is also compared with a naïve batching approach, in which order are batched in an arbitrary sequence until approaching the capacity limitation.

II. Model Description

Say n orders have to be picked, as each order i has a pick profile, consisting of c_{ij}, $j = 1..m$, the number of picks in each location (pick face) j. We assume that the items to be picked have the same physical characteristics (shape, weight, etc.), so they consume the same space in the picking cart. Let C be the maximal number of items that can be picked by the picking cart, so we assume that each order satisfies this constraint, namely, $\sum_{j=1}^{m} c_{ij} \leq C$ for all i.

The proposed approach relies on some analytical results of [11] on sequencing items in a bucket brigade system with two workers. The authors show that, when the items have the same workload distribution along the line, no blockage occurs when workers are sequenced from slowest to fastest. Moreover, if items with different workload distribution are processed, they can be sequenced in any order without blockage, as long as the cumulative workload of the items is the same. These principles are applied here, as the batching process aims in balancing the load, as close as possible to the cart capacity, and at the same time constructing similar picking (workload) distribution (denoted also as *picking profile*) along the picking aisle. In other words, the objective function is twofold; first, to batch orders such that the total number of picks is as close as possible to C (minimizing the number of batches), and second, to build batches with a similar profile. While the first objective is a straight forward and easy to measure, the second objective is less clear, and a new measure has to be developed.

Define an average order profile as a fictitious order, with number of picks in each location j equals to $\bar{c}_j = \left(\sum_{i=1}^{n} c_{ij}\right)/n$. In order to build batches with a pick profile proportional to that of the average order profile, still, with a number of picks as close as possible to C, define the average batch profile as a batch with the number of picks in location j, equals to $\bar{c}_j^b = \bar{c}_j \frac{C}{\bar{c}}$, where $\bar{c} = \sum_{j=1}^{m} \bar{c}_j$ is the average order size.

We first present a non-linear integer programming (NLIP) formulation, and then convert it to an equivalent constraint programming (CP) formulation. Let y_{ik} equals one if order i is assigned to batch k, and zero otherwise. Note that the number of batches is set to some upper bound, UB^b, as some of the batches may be empty. Let z_{kj} be the number of picks in location j for batch k. Last, let z_k to be equal to one if batch k is not empty. The NLIP formulation is given as follows.

NLIP model

$$minimize \sqrt{\sum_{k=1}^{UB^b} \sum_{j=1}^{m} z_k \left(z_{kj} - \bar{c}_j^b\right)^2} \quad (1)$$

subject to

$$\sum_{k=1}^{UB^b} y_{ik} = 1 \qquad i = 1..n \quad (2)$$

$$z_{kj} = \sum_{i=1}^{n} c_{ij} y_{ik} \qquad k = 1..UB^b, j = 1..m \quad (3)$$

$$\sum_{j=1}^{m} z_{kj} \leq C \qquad k = 1..UB^b \quad (4)$$

$$z_k \leq \sum_{i=1}^{n} y_{ik} \qquad k = 1..n \quad (5)$$

$$z_k \geq y_{ik} \qquad i = 1..n, k = 1..UB^b \quad (6)$$

$$z_k \geq z_{k+1} \qquad k = 1..UB^b - 1 \quad (7)$$

$$y_{ik} = (0,1) \quad (8)$$

$$z_{kj} = int. \quad (9)$$

$$z_k = (0,1) \quad (10)$$

The objective function (1) minimizes the differences between each batch profiles and the average batch profile, while ignoring empty batches. Constraint (2) assures that each order is assigned to a single batch. Constraint (3) sums up the number of picks for each batch in each location. The cart capacity constraint is captures in constraint (4). Constraints (5) and (6) set the value of the batch usage variable. Constraint (7) is a symmetry break, as the non-empty batches are sequenced first. Constraints (8-10) are integrality constraints.

Clearly, the NLIP model cannot be solved to optimality via commercial software, due to the non-linear objective function. Instead, we propose the following CP formulation. CP has some advantages over IP, which are given in [12]. Among other advantages, CP is capable of solving non-linear formulations, and it can provide close to optimal solutions for relatively large instances. To this end, we define a new variable, x_i, to hold the batch number to which order i has assigned to. The rest of the notation remain the same.

CP model

$$minimize \sqrt{\sum_{k=1}^{UB^b}\sum_{j=1}^{m} z_k\left(z_{kj} - \bar{c}_j^b\right)^2} \quad (1)$$

subject to

$$z_{kj} = \sum_{i=1}^{n}(x_i = k)c_{ij} \qquad k = 1..UB^b, j = 1..m \quad (11)$$

$$\sum_{j=1}^{m} z_{kj} \le C \qquad k = 1..UB^b \quad (4)$$

$$z_k \le \sum_{i=1}^{n}(x_i = k) \qquad k = 1..UB^b \quad (12)$$

$$z_k \ge (x_i = k) \qquad i = 1..n, k = 1..UB^b \quad (13)$$

$$z_k \ge z_{k+1} \qquad k = 1..UB^b - 1 \quad (7)$$

$$x_i = (0, UB^b) \quad (14)$$

$$z_{kj} = int. \quad (9)$$

$$z_k = (0,1) \quad (10)$$

The objective function remains the same as in the NLIP formulation. Constraint (11) sums up the number of picks for each batch in each location, as the logical condition of assigning order i to batch k multiplies the number of picks of that order in pick face j. The capacity constraint (4) remains the same. In constraints (12) and (13), the value of the batch usage variable is set using the logical condition of the order assignment. The rest of the model remains the same, except the domain definition of x_i, which is belong to the set 1 to UB^b.

III. Prliminary Experimentation

A preliminary experimentation is presented here, for demonstration purposes. Two problems were solved; the small problem contains 24 orders, which are picked from five locations, and the large problem contains 50 orders, which are picked from 12 locations. Items to pick from each location were randomly generated for both instances, and the batching problem was solved via CP. In order to evaluate the batching procedure, the obtained solution was compared via simulation with picking without batching, and arbitrary batch procedure. In the latter, orders were assigned sequentially to batches, up to the maximal allowed capacity. The simulated environment considered two and three pickers, for the small and large problem, respectively. A stochastic forward and backward speed was assumed, using a Normally distributed times, bounded from below by zero, to move between adjacent locations. The expected time of walking backward was assumed two times smaller than walking forward. A stochastics picking time was also assumed, using Normally distributed picking time per item. The parameters of the simulation are given in Table 1.

TABLE I. Pickers rate parameters for the simulation

	Picker 1	Picker 2	Picker 3
Time Forward	Max(N(1.0,0.2),0)	Max(N(1.0,0.2),0)	Max(N(1.0,0.2),0)
Time backward	Max(N(0.5,0.1),0)	Max(N(0.5,0.1),0)	Max(N(0.5,0.1),0)
Picking time	Max(N(1.4,0.2),0)	Max(N(1.1,0.2),0)	Max(N(1.0,0.2),0)

The original structure of the orders of the 24-order, 5-location problem is shown in Fig.2. Each order was randomly generated from a discrete uniform distribution between zero and 15. The batching problem was solved with the CP formulation, for a capacity limit of 200 items. A 5-batch solution was obtained, which is shown in Fig.3. For comparison purposes, a naïve arbitrary batching solution was constructed, simply by sequential accumulation of orders, until meeting the capacity limit. The arbitrary solution, presented in Fig.4, consists of six batches. Along with the excessive batch in the arbitrary solution, one can see that the variability of the batches in each pick location is much smaller in the CP solution.

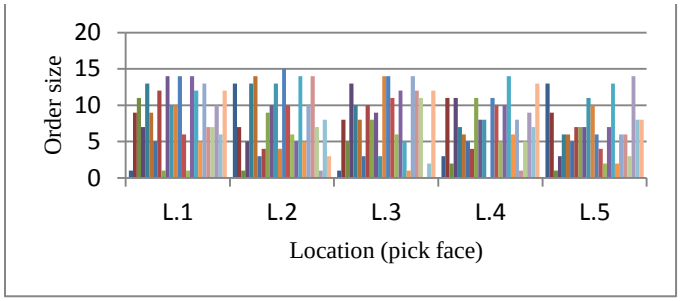

Fig.2. Original orders – no batching (24 orders, 5 locations)

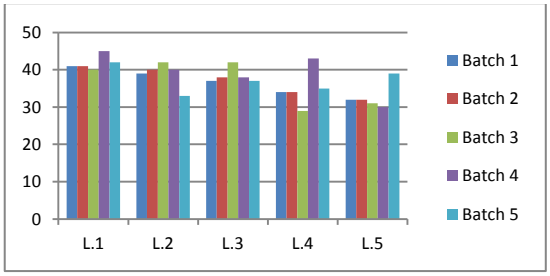

Fig.3. CP batching solution ($\mathbf{C = 200}$)

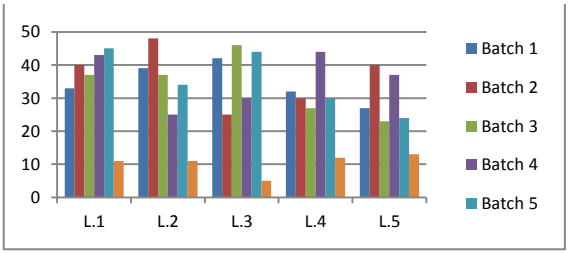

Fig.4. Arbitrary batching solution ($\mathbf{C = 200}$)

In order to overcome the warm-up conditions, each simulation run of the small problem was executed for 120 orders (five times the 24-order data), and 20 replications were taken. The makespan obtained was 3149, 3320 and 3589 for the CP solution, arbitrary batches solution and separate orders (no batching), respectively. Namely, the CP batching solution provides an improvement of 5.4% over the arbitrary batching solution, and 14% over the solution without batching. Note that the mean differences were found statistically significant, with p-value smaller than 1%.

The simulation for the large-scale 50-order, 12-location, problem, showed significant difference, of up to 47%, when compared with the separate orders solution. A much smaller difference (still significant) of up to 12%, was obtained when comparing with the arbitrary batch solution.

REFERENCES

[1] J.A. Tompkins, J.A. White, Y.A. Bozer, and J.M.A. Tanchoco, Facility Planning, John Wiley & Sons, 2010, Technology & Engineering.

[2] M.B.M. De Koster, T. Le-Duc, and K.J. Roodbergen, "Design and control of warehouse order picking: a literature review", European journal of Operational Research, 182(2), 2007, 481-501.

[3] M.B.M. De Koster, E.S. Van der Poort, and M. Wolters, "Efficient orderbatching methods in warehouses", International Journal of Production Research, 37(7), 1999, 1479-1504.

[4] R. Dekker, M.B.M. de Koster, K.J. Roodbergen, and H. van Kalleveen, Improving order-picking response time at Ankor's warehouse, Interfaces, 34(4), 2004, 303-313.

[5] N. Gademann, and S. van de Velde, "Order batching to minimize total travel time in a parallel-aisle warehouse", IIE transactions, 37(1), 2005, 63-75.

[6] S. Hong, A.L. Johnson, and B.A. Peters, "Large-scale order batching in parallel-aisle picking systems", IIE Transactions, , 44(2), 2012a, 88-106.

[7] S. Hong, A.L. Johnson, and B.A. Peters, "Batch picking in narrow-aisle order picking systems with consideration for picker blocking", European Journal of Operational Research, 221(3), 2012b, 557–570.

[8] J.J. Bartholdi and D.D. Eisenstein, "A production line that balances itself", Operations Research, 44 (1), 1996, 21–34.

[9] S. Hong, A.L. Johnson, and B.A. Peters, "Quantifying picker blocking in a bucket brigade order picking system", International Journal of Production Economics, 170, 2015, 862-873.

[10] S. Hong, A.L. Johnson, and B.A. Peters, "Order batching in a bucket brigade order picking system considering picker blocking", Flexible Services and Manufacturing Journal, 2016, 28(3), 425-441.

[11] Y. Bukchin, E. Hanany, and E. Khmelnitsky, "Order sequencing of non-identical items with general work distribution in bucket brigade system", Working paper, 2018.

[12] Y. Bukchin, and T. Raviv, "Constraint programming for solving various assembly line balancing problems", Omega, 78, 2018, 57-68.