An efficient iterated local search algorithm to solve the operational workload imbalance problem

Sarah Vanheusden[1*], Teun van Gils[1], An Caris[1], Katrien Ramaekers[1] and Kris Braekers[1]
[1]Hasselt University, Belgium, * sarah.vanheusden@uhasselt.be

Abstract
To stay competitive and preserve high service levels for customers, the focus of warehouses in today's supply chain is on fast and timely delivery of smaller and more frequent orders. To keep up with competitors, companies accept late orders from customers, which results in additional pressure for order picking operations. Specifically, more orders need to be picked and sorted in shorter and more flexible time windows, which often results in workload peaks during the day. The objective of this study is to formulate and solve the operational workload imbalance problem in parallel zone order picking systems. An iterated local search algorithm is provided to solve the planning problem. Solving the operational workload imbalance problem results in a more stable order picking process and overall productivity improvements for the total warehouse operations.

Keywords: iterated local search, manual order picking, workload balancing, integer programming.

1 Introduction

Warehouses, where products can be stored before the fulfilment of customer orders, play a vital role in the supply chain. To fulfil customer orders, warehouse operations have to satisfy basic requirements such as receiving, storing and retrieving stock keeping units. Considering the four basic warehouse activities (i.e., receiving, storage, order picking and shipping), order picking is the most costly activity. About half of the total warehouse operating costs can be attributed to this process [1]. Order picking, retrieving goods from storage or buffer areas to fulfil incoming customer orders, is labour intensive whenever it is performed manually, and very capital intensive if automated warehouse systems are used. Despite technological advances in automated warehouse systems, the most dominant order picking systems in practice still rely on human operators. Human operators yield more flexibility to the order picking process due to the combination of their motor and cognitive skills [2]. This flexibility is needed to preserve high service levels in times of a changing customer behaviour. Trends such as shortened product life cycles, e-commerce, greater product variety and the trend of accepting late orders from customers, result in extra pressure for order picking operations. More orders need to be picked and sorted in shorter and more flexible time windows. Warehouse managers therefore experience difficulties in balancing the workload of the order pickers on a daily basis, resulting in peaks of workload during the day [3].

The problem addressed in this paper originates from a large international B2B warehouse located in Belgium. The warehouse is responsible for the storage and distribution of automotive spare parts. Spare part warehouses are characterized by customer orders that can be grouped based on their destination. An order set refers to a group of order lines with a common destination that is picked in a single zone. All order lines of a common destination from all pick zones are referred to as the order lines from a shipping truck. Deadlines of order lines are determined by the shipping truck and resulting schedule of shipping trucks (i.e. shipping schedule). Each shipping truck can consist of multiple order sets (i.e., a single order set for each order picking zone). The assignment of order sets to shipping trucks as well as the shipping schedule are assumed to be fixed at the operational level. This fixed shipping schedule often results in workload peaks during the day, as order patterns vary across customers (e.g., varying number of orders and customers, varying order time and resulting available time to pick orders) [3].

Balancing the workload in an order picking system can be addressed from different perspectives. While most papers that cover the issue of workload imbalance, start at a strategic or tactical level, the emphasis of this paper is on the operational level with the aim of avoiding peaks in the number of order lines to be picked during the day. The objective function aims to minimize the variance of planned order lines over all time slots, for each pick zone. [3]. Solving instances of realistic size to optimality in a reasonable amount of computation time does not seem feasible due to the complex nature of the operational workload imbalance problem (OWIP). The objective of this paper is to develop an iterated local search algorithm to solve OWIP in a parallel zoned manual order picking system. This study contributes to the current state-of-the-art by providing a more stable order picking process and overall productivity improvements for the total warehouse operations.

2 Problem formulation

This section describes the operational workload imbalance problem as introduced in [3]. The notation outlined in Table 1 is used. A mixed integer programming (MIP) model is developed to formulate OWIP.

Table 1: Sets, parameters and decision variables

Sets	
ι	set of time slots with time slot $i = 1, 2, \ldots, I$
κ	set of pick zones with $k = 1, 2, \ldots, K$
λ	set of shipping trucks with $l = 1, 2, \ldots, L$
Parameters	
a_{kl}	number of order lines of shipping truck l in zone k
Δt^{max}	maximum difference in number of time slots that is allowed for planning order sets of a shipping truck over different zones
t_l^r	release time slot of shipping truck l
t_l^d	deadline time slot of shipping truck l
δ	split order set factor for splitting large order sets over multiple consecutive time slots
μ_k	mean number of order lines per time slot in zone k
Decision variables	
X_{ikl}	binary variable which is equal to 1 if and only if order set $(k; l)$ is planned in time slot i

Objective function

$$\min \sum_{k \in \kappa} \frac{1}{I} \sum_{i \in \iota} \left(\sum_{l \in \lambda} a_{kl} X_{ikl} - \mu_k \right)^2 \tag{1}$$

Subject to

$$\sum_{\substack{i \in \iota \\ i < t_l^r \\ i > t_l^d}} X_{ikl} = 0 \qquad \forall k \in \kappa \qquad \forall l \in \lambda \tag{2}$$

$$\sum_{i \in \iota} X_{ikl} = 1 \qquad \forall k \in \kappa \qquad \forall l \in \lambda \tag{3}$$

$$\sum_{i \in \iota} i\left(X_{ikl} - X_{ik'l}\right) \leq \Delta t^{max} \qquad \forall k \in \kappa \qquad \forall k' \in \kappa \backslash \{k\} \qquad \forall l \in \lambda \tag{4}$$

Objective function 1 minimizes the variance of planned order lines over all time slots, for each pick zone. Constraints 2 indicate that an order set (i.e., the combination of $\{k; l\}$) can only be scheduled after the release time of a shipping truck and before the pick deadline of the corresponding shipping truck. Assigning each order set to a single time slot is the result of constraints 3. Constraints 4 incorporate the maximum difference in time slots for planning order lines of a certain shipping truck over different zones. This difference in time slots cannot exceed a threshold parameter Δt^{max} in order to prevent the model creating huge buffers in the staging area.

In addition to the above constraints, the model includes an extra parameter δ in case of large order sets. To avoid the model planning an large order set in a single time slot, which will be infeasible to pick in practice, the split order set factor δ is defined as a fraction of the mean number of order lines per time slot in zone k. Each order set $\{k; l\}$ is split into two if the following equation is met: $a_{kl} > (1 + \delta)\mu_k, \forall k \in \kappa$ and $\forall l \in \lambda$. By means of the size of δ, order sets are split into two if an order set of shipping truck l is greater than $(1 + \delta)$ times μ_k in order to facilitate balancing over the different time slots. Furthermore, the split order sets must be scheduled in consecutive time slots, which results in an extra set of equations:

$$\left| \sum_{i \in \iota} i\left(X_{ikl^1} - X_{ikl^2}\right) \right| \leq 1 \qquad \forall\left(l^1; l^2\right) \in \lambda' \qquad \forall k \in \kappa \tag{5}$$

with $\left(l^1; l^2\right) \in \lambda'$ the set containing the split order sets, and l^1 and l^2 the first and second part of split order sets, respectively.

3 Methods

Due to the complex nature of OWIP, solving instances of realistic size to optimality in a reasonable amount of computation time does not seem feasible. Therefore, an iterated local search (ILS) algorithm is introduced to solve the operational workload imbalance problem. First, an initial solution is created in which orders of all shipping trucks are assigned to the deadline time slot $\left(t_l^d\right)$. By means of a local search procedure, the initial solution is improved. The local search procedure consists of reassigning an order set to another time slot (i.e., shift), exchanging two order sets of different time slots (i.e., swap) and an ejection chain. Moves are repeated until no further improvement in the objective function value is possible. This local search procedure results in a local optimum. To escape from this local optimum, a large and random change is performed to the currently best found solution. The perturbation is followed by the local search procedure to reach a new local optimum. If an iteration of perturbation and local search results in an improved solution with respect to the variance, the solution is accepted as new best solution. These steps are repeated until $1,000$ consecutive iterations without improvement.

Table 2: Warehouse parameter values

Warehouse parameter		**Parameter value**
$U(u_1; u_2)$	uniform distribution with u_1 and u_2 the lower and upper bound	
I	number of time slots	24 time slots
a_l	number of order lines of shipping truck l	$N(\mu_a; \sigma_a)$
a_{kl}	number of order lines of shipping truck l in zone k	$a_l\left(\frac{1}{K} + U(-\gamma; \gamma)\right)$
μ_a	mean number of order lines of a shipping truck	175 order lines
γ	split shipping truck factor that defines the distribution of order lines across pick zones	0.05
δ	split order set factor	0.25
t_l^r	release time slot of orders in shipping truck l	$U\left(1; I\right)$
t_l^p	available number of time slots to pick shipping truck l	$U(\mu_{t^p} - \sigma_{t^p}; \mu_{t^p} + \sigma_{t^p})$
σ_{t^p}	variation in number of time slots between t^r and t^d for each l	2 time slots
t_l^d	deadline time slot of orders in shipping truck l	$\min\left(I; t_l^r + t_l^p\right)$

To assess the performance of the proposed ILS algorithm, a series of experiments is performed. Warehouse parameters are outlined in Table 2. In the experiments, five factors are considered as summarised in Table 3.

Table 3: Experimental factor setting

Factor		**Factor levels**		
L	number of shipping trucks	(1) 100 trucks	(2) 150 trucks	(3) 200 trucks
K	number of pick zones	(1) 1 pick zone	(2) 2 pick zones	(3) 3 pick zones
Δt^{max}	maximum difference in t_i for planning orders of l in each k	(1) 2 time slots	(2) 4 time slots	(3) 6 time slots
σ_a	variation in number of order lines for each l:	(1) 5 order lines	(2) 10 order lines	(3) 15 order lines
μ_{t_p}	mean number of time slots between t^r and t^d for each l	(1) 1 time slot	(2) 2 time slots	(3) 3 time slots

4 Results

The algorithm is implemented in `C++`. To solve the MIP formulation, `ILOG Cplex 12.7` is used with a runtime limit of 5 h. As even the small instances could not be solved to optimality, due to the quadratic objective function, the imbalance is approximated by minimizing the range of OWIP in each order picking zone as follows: $\min \sum_{k \in \kappa} \left(A_k^{max} - A_k^{min}\right)$. A_k^{max} and A_k^{min} are defined as the maximum and minimum number of planned order lines over all time intervals in each order pick zone, respectively. It can be proven that the range of a given solution is minimal if the variance of a solution is minimal. Consequently, minimising the variance of OWIP by the heuristic algorithm and calculating the range of the resulting solution allows us to evaluate the performance of the ILS algorithm with respect to the optimal solution.

As we are currently developing the ILS algorithm, this section presents some preliminary results of the performance of the ILS algorithm. The ILS algorithm is tested on small problem instances with an experimental factor setting as shown in Table 3. All 243 factor combinations are replicated 5 times, which leads to a total of 1, 215 instances. Results of solving the revised MIP model with `Cplex` show that 930 instances have been solved to optimality within the 5 h limit (76.5 %). The optimality gap of the remaining 285 instances varies between 0.02 % and 86.63 % with an average gap of 8.68 %. The number of instances that cannot be solved to optimality by `Cplex` increase with larger values of factors L, K and μ_{t_p}.

The 930 instances that have been solved to optimality are compared to the results of the ILS algorithm in terms of their performance in objective function value (range). The difference in the range between the optimal solution and the ILS algorithm is on average 3.11 (0.87 %). For all 1, 215 instances, the average runtime of `Cplex` is compared to the average runtime of the ILS algorithm. The average runtime of solving the MIP formulation is about 103 times higher than the average runtime of the ILS algorithm. The average runtime of the optimal solution amounts 4, 386.1 s, while the ILS algorithm only takes 42.6 s on average. This large difference in runtime has important implications for the application possibilities of OWIP. The intention of the developed model, is its usage as a simulation tool to plan order sets more accurately during the day, in this way, avoiding peaks in workload. The model can be used to support warehouse managers and supervisors in their daily planning activities. Long run times for OWIP means a smaller application potential in practice. The proposed ILS algorithm can serve as an alternative for the exact algorithm, providing fast and accurate results whenever it is used as a supporting tool in practice. However, the algorithm needs further testing on realistic instances to be valuable in practice.

5 Conclusion

Short time periods for picking customer orders cause peaks in workload during the day, resulting in extra pressure for order pickers. Only approaches for long-term balancing have been introduced in literature. Practitioners are searching for a solution to balance the workload for every hour of the day, to increase the utilization of pickers and decrease the probability of missing shipping deadlines. Results show that the proposed ILS algorithm is able to balance the workload during the day for small instances. To make our research more valuable to practice, additional experiments are required: the algorithm should be tested for realistic problems to show the benefits in practice. The proposed algorithm can be used by warehouse supervisors as a decision support tool to plan order sets more accurately during the day, in this way, avoiding peaks in workload. Additionally, the algorithm can serve as an advisory tool for managers to start negotiations in changes in cut-off times for customer order entry and shipping schedules to further reduce workload imbalances.

Acknowledgements

This work is supported by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office (research project COMEX, Combinatorial Optimization: Metaheuristics & Exact Methods). The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government – department EWI.

References

[1] T. Van Gils, K. Ramaekers, A. Caris, and R. De Koster. Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review. *European Journal of Operational Research (in press)*, 000: 1–15, 2018.

[2] E. H. Grosse, C. H. Glock, M. Y. Jaber, and W. P. Neumann. Incorporating human factors in order picking planning models: framework and research opportunities. *International Journal of Production Research*, 53(3): 695–717, 2015.

[3] S. Vanheusden, T. van Gils, K. Ramaekers, A. Caris. Reducing workload imbalance in parallel zone order picking systems. *Proceedings of the International Conference on Harbour, Maritime & Multimodal Logistics Modelling and Simulation, HMS 2017*, Barcelona, Spain, September 18–20, 2017.