A HYBRID HEURISTIC FOR ORDER-PICKING AND ROUTE PLANNING IN WAREHOUSES

Jerzy DUDA, Marek KARKULA

AGH University of Science and Technology, Faculty of Management, Cracow, Poland, EU,
jduda@zarz.agh.edu.pl, mkarkula@zarz.agh.edu.pl

Abstract

The organization of the order picking process has the greatest impact on the efficiency of the warehouse or distribution center, and thus on the supply chain efficiency. From the moment of accepting the customer's order through its completion to the time of shipment, there are many possibilities of making mistakes and errors both in terms of accuracy and completeness, as well as time-wasting. The paper presents the problem of order-picking in a typical warehouse in which items must be collected manually by the workers available in a given period of time (e.g. working shift). The goal is to develop a plan that minimizes the number of pickers that must be involved in picking orders in a given section of a warehouse and simultaneously optimizes the distance covered by the pickers. Mixed integer programming model for the order-picking and route planning problem is formulated and the heuristic based on genetic algorithm and TSP r-opt technique is proposed to solve it. Proposed genetic algorithm uses a crossover dedicated to the special representation of the solution coding order-picking plan and four different mutations allowing for extensive exploration of solution space resulting in a high quality, reliable order-picking plan. The quality of the proposed solution is evaluated against the data coming from one of the real warehouses. It was possible to achieve a significant reduction in the number of pickers necessary to collect all items as well as the total distance covered by the pickers.

Keywords: Order-picking, route planning, warehouse, optimization, heuristic

1. INTRODUCTION

The problem of order-picking is defined as the collection of items from their location in the warehouse in order to satisfy the demand from internal or external customers. Petersen and Schmenner [1] also distinguish two types of the order-picking problem in which human resources (pickers) are involved:

- items-to-picker, in which the items are automatically delivered to the picker at the point of collection and dispatching of items,
- picker-to-items, in which an employee (picker) travels to different locations in order to pick up the needed items.

In many warehouses we should deal with the second type of the order-packing problem. Despite the increasing automation of the process, according to the literature, still around 80 % warehouses in Western Europe use picker-to-items system operated manually [2]. Various studies also proves how crucial is the problem to the warehouse operations management - the costs of this process account for 50-65 % of the total operating costs [2-5]. Despite these facts, optimization of the order-picking process is relatively poorly recognized in the literature, if we compare it to, for example, to job scheduling problem or even warehouse location problem. Wäscher [6] has reviewed various aspects and various methods that can be applied to a picker-to-commodity problem. Among the most important factors that influence the planning of the pickers work he enumerated:

- allocation of items to warehouse location - fixed area for items or random placement of items (depending on deliveries to the warehouse),
- transformation of customer orders into picking lists (consolidation policy) - execution of individual orders or consolidation of smaller orders into picking lists,

- routes of pickers moving through the warehouse (routing policy) - arrangement of paths in the warehouse on which a picker can travel and acceptable directions of movement (S-shape, return, largest-gap, mixed strategies).

In this paper we focus on a warehouse which uses random placement strategy of items, implements individual picking lists (consolidation takes place earlier) and a routing policy that requires a picker to return to the main path (return strategy). This means that the analyzed problem is characterized by the highest possible complexity and such requires development of dedicated optimization algorithms.

The goal of the paper is to develop an optimization model of the order-picking problem and to build a method able to provide a solution to this problem in an acceptable time and with an acceptable quality. The method has been built basing on a genetic algorithm metaheuristic. The algorithm is hybridized with a TSP optimization in order to optimize both number of engaged pickers and total distance the pickers have to travel.

2. PROBLEM DEFINITION

The main goal is to develop the picking plan that minimizes the number of pickers engaged in picking process in a given area and simultaneously to optimize the route they have to travel when picking orders from completion list. The main assumption is that items picking for all orders in each completion list must be finished before deadline (beginning of delivery to a customer).The mixed integer programming (MIP) model can be presented as follows:

Data

d_i - due date (time) for item i
s_i - the earliest possible time for picking item i (availability time),
e_p - performance of picker p (expressed e.g. as the number lines per hour),
a_{pk} - 1, if picker p is available in location k; 0 otherwise,
g_{ik} - 1, if item i is located in location k; 0 otherwise,
b_k - maximum number of pickers that can work in location k,
$\delta(i,j)$ - distance between locations of items i and j that follows routing strategy,
P - number of pickers available,
T - number of time windows,
I - number of items to be collected,
J - number of items preceding picking of item i,
K - number of different locations in the store.

Decision variables:

x_{ipt} - 1, if picker p is due to pick item i in time windows t; 0 otherwise,
r_i - time window in which item i is due to be picked,
m_{ijpt} - 1, if picker p after collecting item i collects item j in time t.

Objective function:

$$w_1 \sum_{p=1}^{P} \max_{i=1..I,t=1..T} x_{ipt} + w_2 \sum_{p=1}^{P} \sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{j=1}^{J} x_{ipt} l_{ijpt} \delta(i,j) \rightarrow min \quad (1)$$

Constraints:

$$\sum_{i=1}^{I} x_{ipt} \leq e_p \,, p = 1, \dots, P, t = 1, \dots, T \quad (2)$$

$$\sum_{t=1}^{T}\sum_{p=1}^{P} x_{ipt} = 1\,, i = 1, \ldots, I \tag{3}$$

$$(x_{ipt} \cdot g_{ik}) \geq a_{pk}, i = 1, \ldots, I; t = 1, \ldots, T; p = 1, \ldots, P; k = 1, \ldots, K \tag{4}$$

$$r_i \leq d_i, i = 1, \ldots, I \tag{5}$$

$$r_i \geq s_i, i = 1, \ldots, I \tag{6}$$

$$\sum_{p=1}^{P} x_{ipt} \cdot g_{ik} \leq b_k\,, i = 1, \ldots, I; t = 1, \ldots, T; k = 1, \ldots, K \tag{7}$$

Goal function (1) optimizes a weighted sum of the pickers engaged in the picking process in a given planning horizon and the total length of the routes the pickers have to travel in order to pick all items. Constraint (2) ensures that picker performance is not exceeded. According to constraint (3) only one picker can pick a single item *i* in a time window *t*. The location in which picker *p* can operate is limited by constraint (4). Constraint (5) says that all items must be picked before their due dates and simultaneously not earlier than the time of their availability - constraint (6). Finally, constraint (7) limits the number of pickers that can work in the same location in the same time window.

3. PROPOSED SOLUTION

Initially we tried to employ CPLEX solver, one of the most efficient MIP solvers basing on branch and cut method and many preprocessing and probing techniques as well as diving and neighborhood search heuristics enabling speed up of the optimization process. However, after series of experiments, we found that for the problem size with only 5 pickers an a time window granularity of 10 minutes, CPLEX solver required more than 15 minutes to achieve the optimized order-picking plan. This time tends to increase in a nonlinear way, when the number of pickers grows and time windows decrease. Such method cannot be implemented in the majority of real industrial conditions.

Considering the above, we had to develop a heuristic able to optimize larger instances of the presented problem. Basing on the analysis of literature and our own experience, we have applied a genetic algorithm (GA) metaheuristic to solve it.

Genetic algorithm-based approach has been used among others by Chang et al. [7], however they used it to optimize the problem of orders-picking in an automated warehouse where the goal functions were minimization of total distance covered and total time necessary for picking all the items. In fact the problem was a version of a classical TSP problem, which is well recognized in literature and there are many genetic operators defined to solve it. Zhang and Liu [6] have analyzed similar problem with total distance minimization and also solved it using genetic algorithm.

The main procedure of the genetic algorithm solving the problem presented in section 2 is shown in **Figure 1**. It follows the standard GA scheme presented e.g. in [8]. However, contrary to the TSP-like problem optimized by Chang et. al. [7] or Zhang and Liu [9], we have to develop specific representation of the problem, saying which picker will be engaged in picking which item and in which time window (variable x_{ipt}). These representations (GA chromosome) consist of two vectors, both of size *I* (number of items to pick). The first vector ***p*** represents picker's number that will pick an item 1…*I*, while the second vector ***t*** represents the ending time for the window in which the picker will pick the item 1…*I*. **Table 1** shows an example of such representation for the 10 first items and 3 pickers.

Table 1 Representation of the solution for first 10 items.

i	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
p_i	1	2	1	2	1	2	2	3	1	3
t_i	06:30	07:30	09:00	08:00	06:00	08:00	07:00	08:30	08:00	07:30

The starting population is initialized in a random way, i.e. picker numbers are drawn from the range 1,...,*P*, and the time window is drawn from the time windows range in which the chosen picker is available. Evaluation phases first checks if all the constraints (2)-(7) are satisfied, and if not, a penalty is given for each constraint violation. Next the first part fitness (goal) function (1) is evaluated, i.e. the number of pickers engaged in the order-picking process. After this stage for only the best solution found concerning the first part of the goal the TSP optimization is fired to optimize the order of locations a given picker has to visit. The optimization procedure, event as simple as 2-opt [10, 11] that we have applied in our experiments takes much more time, than evaluation of the first fitness function, so applying it only for the best solution saves a lot of time.

```
Pop←Initialize()
foreach x in Pop:
x_f ←Evaluate()
best ←FindBest()
best ←OptimizeTSP()
foreach generation in 1..max_generation:
        Pop ←TournamentSelection()
        foreach x in Pop step 2:
                if rand()<p_cros
                        x ←Crossover(x,x_next)
        foreach x in NewPop:
                if rand()<p_mut1
                        x ← ChangePeriodMutation()
                if rand()<p_mut2
                        x ← SwapPeriodMutation()
                if rand()<p_mut3
                        x ← ChangePickerMutation()
                if rand()<p_mut4
                        x ← SwapPickerMutation()
                if rand()<p_mut5
                        x ← DecreasePickersMutation()
        foreach x in Pop:
        x_f ←Evaluate()
        best ←FindBest()
        best ←OptimizeTSP()
return best
```

Figure 1 The main procedure of genetic algorithm

Algorithm continues its execution for the number of generations set in *max_generation* parameter. In each generation first the solutions from old population are chosen to a new one on the basis of tournament selection (from two randomly chosen solutions the solution with better goal function values is taken). For each two consecutive solutions (parents) crossover operator is applied with a probability of p_{cros}. Standard two point crossover has been used [8]. This operator simply exchanges the values in the chromosomes after a randomly chosen gene (head and tail are swapped between two parents to produce their children). Such a simple, standard crossover operator can be used, because after exchanging values of $\boldsymbol{p}$ vector along with the corresponding $\boldsymbol{t}$ vector values, the resulting solution will still be valid.

Next, each solution in the new population can be mutated with a relatively small probability. Five mutation operators have been develop in order to control the search process in a very precise way. The mutation works as follows:

- *ChangePeriod* - changes a randomly chosen time window to another one within the acceptable range, i.e. between the availability time of the item i to be collected and the collection time required.
- *ChangePicker* - changes a randomly chosen picker number to a different number.
- *SwapPeriod* - changes time windows for the two randomly chosen items. The availibilty of the item is checked and if the time is smaller than it, the time in changed to a correct value, if the time is greater than the required collection time, this time is adjusted accordingly.
- *SwapPicker* - swaps the picker numbers for the two randomly drawn items.
- *DecreasePickers* - the most advanced mutation that can potentially reduce the number of pickers involved in collecting items in a given time window. The mutation works in such a way that the picker that has the smallest number of times to collect is selected. The selected picker is then change for a different picker number.

4. COMPUTATIONAL EXPERIMENTS

The tests allowing for the evaluation of the proposed solution have been carried on the data taken from the one big warehouse with the automotive products and spare parts located in Poland. Basing on the real data we have prepared two sets of times to be picked for 8 hours. First one consisted of 100 items and 10 available pickers with the planning horizon of 8 hours, while the second one consisted of 300 items and 20 pickers with the planning horizon of 16 hours. **Table 2** presents the results for the set with 100 items for different window time granularity.

Table 2 Results for the set with 100 items and 10 pickers [own study]

Time window size	Number of time windows	Number of pickers	Distance improvement	Execution time
1 h	8	2	15 %	<1s
15 min	32	3	12 %	3s
5 min	96	4	8 %	6s
1 min	480	4	8 %	24s

As we can see in **Table 2**, even for the tightest time window with 1 minute the algorithm is able to return solution after the time no more than 30 seconds, which is quite acceptable not only for planning the routes for pickers, but also for replanning in the case something went not as it has been planned and the plan needs to be adjusted to match the current situation in the warehouse.

Number of pickers has been reduced to 2-4, depending on the time window. It can be observed that the smallest time window is, more pickers have to be engaged in the picking process. This is because the performance of the pickers is checked more precisely if the window is smaller, as we use an average normative performance. Improvement of the total distance the pickers have to travel tends to decrease along with the number of pickers. This is because the gain for optimization of longer routes is on average higher than from optimization of shorter routes. Additional pickers in our case have to collect only one or two items, so their route cannot be optimized at all.

For the set with 300 items the results were analogous, however, the running time for the time window of 1 minute increased to 7 minutes, which still seems to be acceptable for majority of the real planning scenarios.

Time for the 1 hour time window was still very short - 7 minutes. The average gain from the pickers routes optimization raised by 21 %.

5. CONCLUSIONS

A hybrid genetic algorithm with TSP optimization method proved to achieve good results and is able to plan ordering-picking process in a warehouse in a relatively quick time, so the proposed method can be applied also for replaning purposes, what makes it very competitive when compared with simple methods or semi-manual planning, traditionally used. The performance of the proposed solution depends strongly on the size of the time window that is used for planning. It is up to the decision maker how small it should be. It is very important, as smaller time windows contribute to the increase of the number of workers involved in the picking process and to the decrease of the average reduction of the routes the pickers have to travel.

Further study in this area will be focused on the application of better TSP optimization algorithms than 2-opt and reduction of overall running time for large numbers of items to be picked and very precise time windows.

ACKNOWLEDGEMENTS

This study was conducted under a research project funded by a statutory grant of the AGH University of Science and Technology in Krakow for maintaining research potential.

REFERENCES

[1] PETERSEN II, Charles G., SCHMENNER, Roger W. An Evaluation of Routing and Volume-based Storage Policies in an Order Picking Operation. *Decision Sciences*. 2007. vol. 30, pp. 481-501.

[2] ŽULJ, Ivan, GLOCK, Christoph H., GROSSE, Eric H., SCHNEIDER Michael. Picker routing and storage-assignment strategies for precedence-constrained order picking. *Computers & Industrial Engineering*. 2018. vol. 123, pp. 338-347.

[3] HABAZIN, Josip, GLASNOVIĆ, Antonia , BAJO, Ivona. Order Picking Process in Warehouse: Case Study of Dairy Industry in Croatia. *Promet - Traffic & Transportation.* 2017. vol. 29 no. 1, pp. 57-65.

[4] PANSART, Lucie , CATUSSE, Nicolas, CAMBAZARD, Hadrien. Exact algorithms for the order picking problem. *Computers & Operations Research*. 2018. vol. 100, pp. 117-127.

[5] SCHOLZ, Andre, WÄSCHER, Gerhard. Order Batching and Picker Routing in manual order picking systems: the benefits of integrated routing. *Central European Journal of Operations Research*. 2017. vol. 25 no. 2, pp. 491-520.

[6] WÄSCHER, Gerhard. Order Picking: A Survey of Planning Problems and Methods. In: DYCKHOFF, Harald, LACKES, Richard, REESE, Joachim (eds.). *Supply Chain Management and Reverse Logistics*. Springer, 2004, Berlin, Heidelberg, pp. 323-347.

[7] CHANG, Fa-liang , LIU, Zeng-xiao, XIN, Zheng, LIU, Dong-dong. Research on order picking optimization problem of automated warehouse. *Systems Engineering - Theory & Practice*. 2007. vol. 27 no. 2, pp. 139-143.

[8] MICHALEWICZ, Zbigniew. *Genetic algorithms + data structures = evolution programs (3rd ed.)*, Springer-Verlag, 1996, Berlin, Heidelberg.

[9] ZHANG, Haijun, LIU, Bingwu. A New Genetic Algorithm for Order-Picking of Irregular Warehouse. In: International Conference on Environmental Science and Information Application Technology, Wuhan, 2009, pp. 121-124.

[10] JOHNSON, David S., McGEOCH, Lyle A. The Traveling Salesman Problem: A Case Study in Local Optimization. In: AARTS, Emile H.L. and LENSTRA, Jan K. (eds.). *Local Search in Combinatorial Optimization*. John Wiley and Sons, 1997, London, pp. 215-310.

[11] NILSSON, Christian. *Heuristics for the Traveling Salesman Problem*. 2003, Linköping University, pp.1-6.