

# Innovative Modular Approach based on Vehicle Routing Problem and Ant Colony Optimization for Order Splitting in Real Warehouses

Emir Žunić<sup>1,3</sup>, Sead Delalić<sup>2,1</sup>, Zlatan Tucaković<sup>3,1</sup>, Kerim Hodžić<sup>3,1</sup>, Admir Beširević<sup>2,1</sup>

<sup>1</sup>Info Studio d.o.o., Sarajevo, Bosnia and Herzegovina

<sup>2</sup>Faculty of Science at University of Sarajevo, Bosnia and Herzegovina

<sup>3</sup>Faculty of Electrical Engineering at University of Sarajevo, Bosnia and Herzegovina

Emails: emir.zunic@infostudio.ba, {delalic.sead, besirevic.admir}@pmf.unsa.ba, {zlatan.tucakovic, kerim.hodzic}@etf.unsa.ba

**Abstract**—A crucial part to any warehouse workflow is the process of order picking. Orders can significantly vary in the number of items, mass, volume and the total path needed to collect all the items. Some orders can be picked by just one worker, while others are required to be split up and shrunk down, so that they can be assigned to multiple workers. This paper describes the complete process of optimal order splitting. The process consists of evaluating if a given order requires to be split, determining the number of orders it needs to be split into, assigning items for every worker and optimizing the order picking routes. The complete order splitting process can be used both with and without the logistic data (mass and volume), but having logistic data improves the accuracy. Final step of the algorithm is reduction to Vehicle Routing Problem where the total number of vehicles is known beforehand. The process described in this paper is implemented in some of the largest warehouses in Bosnia and Herzegovina.

## I. INTRODUCTION

THE workflow of large distribution companies consists of acquiring items from the manufacturer and storing them in large warehouses. When orders are made, items are delivered and distributed to customers. Distribution companies can deliver thousands of orders daily to various types of customers: smaller markets, shopping malls, restaurants etc. Each order has to be collected from the warehouse.

The process of collecting orders is usually called the order picking and it takes a considerable part of the warehouse workflow. The order picking is the most demanding process and it takes approximately 55% of the warehouse operating cost [1]. It is usually done by warehouse workers who are walking between the racks and collecting items in a trolley or a cart. In some cases items are picked up with a forklift. The picking route is determined before the process of collecting items begins. Each process is controlled by the Warehouse Management System (WMS) [2].

The process of finding optimal route is equivalent to the standard Traveling Salesman Problem (TSP) with additional constraints such as picking heavy objects first and placing them at the bottom cart or placing items last because they have special temperature requirements. Many heuristic algorithms

are given to solve this problem, so the process of order picking can be optimized.

Orders can contain between one and more than a thousand of distinct items and in that case, the picking process can be a challenging task. The optimal route can have the length of several kilometers and the mass and the volume of picked items can exceed the capacity one worker can handle. In that case, the worker must pause the picking process, go to the unloading location and unload items, prior to resuming the order picking process. The problem with this solution are common cases in which the unloading location is far from the current picking location, so the route length is significantly increased. However, the order splitting is imposed as an intuitive and natural solution. The process of order splitting can improve order picking process in general, but the order splitting has to be done in a way that the operating costs are reduced. This paper presents the innovative approach for order splitting implemented in real warehouses.

The main goal is to split the order to a number of smaller and balanced orders. This can also reduce a psychological burden on the worker, and increase the possibility of establishing a better balance between tasks among the workers. The orders can be balanced by the total route length, mass, volume or other parameters. The mass, volume or other logistic data are not always available or accurate. The process of order splitting needs to be adjusted in a way that works well in both cases.

The process of order splitting consists of several steps. The first step is to determine the number of smaller orders which the main will be divided into. Some orders do not need to be divided, because they do not require the transportation over a large distance and do not contain volume and mass that overload one worker. The second step is the process of dividing items to smaller orders and the process of route optimization for each order.

The algorithm described in this paper is implemented in some of the largest warehouses in Bosnia and Herzegovina. These warehouses deliver in real-time, so the orders are picked as soon as possible. In warehouses where the 48-hours delivery

system is used, orders are known at the beginning of the day. This means that everything can be planned before the first order is collected. In these cases, order batching can improve the picking process. The order batching is the process of merging smaller orders into a larger one, so the worker can simultaneously pick all of them. The batching is not possible when the orders are not known before, which is the case in real-time delivery systems. The algorithm presented in this paper can be of great importance for advanced order splitting in warehouses with real-time delivery system and can be used in warehouses with non-standard layouts.

In the first section, the short introduction and motivation for this paper is given. The literature review is given in the second part. In the third section, the complete process of order splitting is described. Results are given in the fourth section, as well as the comparison of the proposed algorithm and other popular approaches. Last section gives the short summary and the future work ideas.

## II. RELATED WORK

The algorithm implemented in the paper is an upgrade of the smart Warehouse Management System described in paper [3]. Order splitting requires distance calculation between any two positions in a warehouse. System uses distance calculation inside a warehouse using a dynamical algorithm for traditional warehouse layouts implemented in paper [4], as well as for nontraditional layouts, described in paper [5]. Order picking route can be additionally reduced if items are placed according to their ordering amount and frequency. More frequent items are at better locations, i.e. closer to the exits. This is implemented as algorithm and described in [6]. Orders are picked faster if workers have enough items in warehouse picking zone and they do not have to use forklifts. Algorithm that predicts item sales based on sales history is described in [7].

The algorithm proposed in this paper uses similarity of order splitting and Vehicle Routing Problem (VRP), while criteria is that routes should have a balanced load. Task of finding the optimal solution for VRP is extremely hard, but sub-optimal solutions can be found using some heuristic or metaheuristic algorithms. Variant of VRP with balanced load routes is presented in [8]. Tabu search is one heuristic method that gave some of the best solutions in this field. In [9] author described approximate methods based on descent, hybrid simulated annealing/tabu search with various search methods. Algorithm computation time is optimised with specific data tables organisation. Another tabu search based on heuristic is TABURoute and it is described in paper [10]. It is based on a row of adjacent solutions acquired by generalized insertion procedure which is replacing one vertex from route to route and recalculating route lengths. Customer window constraints represent period of time in which customer is available to receive transported goods. Those constraints are common in real life situations and they can make VRP problem solving more complicated. They are usually referred as VRPTW (Vehicle Routing Problem with Time Windows) in literature.

Mathematical model and optimization-based algorithms using route length minimization as criteria is described in paper [11]. Operational research methods can also help solving this type of problems. In paper [12] optimization algorithm based on linear and dynamic programming with feasible column generation is proposed. Linear programming solution was used as lower bound for branch-and-bound algorithm used for solving integer set partitioning formulation. This concept delivered optimal results for a hundred customers VRP problems. Algorithm and concept described in this paper uses Ant Colony Optimization for solving VRP. Ant Colony Optimization is introduced and described in paper [13]. That algorithm gives very good sub-optimal solution, within 1% of known optimal solutions, and uses multiple ant colonies. Another approach for order splitting based on Poisson processes is given in [14], assuming that arrival process is distributed by Poisson distribution.

## III. CASE STUDY

In this section, the complete algorithm for order splitting is described. The main goal of the algorithm is to optimally split the given order to smaller ones.

The proposed algorithm consists of three main steps:

- Collecting data and distances calculating
- Determining the number of smaller orders
- Order splitting

### A. Collecting data and distance calculating

The process of collecting data begins with determining locations in warehouse from which the items will be collected. In the concept of smart warehouse the items with the earliest expiration date are picked, so the locations are easy to determine. As mentioned before, many warehouses do not have complete logistic data. Often, the mass or the volume of the items are missing. In addition, in many practical cases, the data are inaccurate (inconsistent unit of measure or typographical error). Therefore, the proposed algorithm for order splitting must work smoothly in situations with inaccessible logistic data. The following data are observed: item location, number of ordered units, mass and volume (if known). Before the next step of algorithm, individual item records are aggregated for each location. The items will be referred as locations in further discussion. The distance between each two locations are calculated. Papers [4] and [5] propose the algorithm for distance calculation in warehouses of various layouts. The mentioned algorithm is used in this implementation.

### B. Determining the number of smaller orders

The process of calculating number of smaller orders ( $k$ ) consists of checking the size of the order and using known data for the split. As size metric total number of different items, the total number of items, the total mass ( $m_{tot}$ ) or the total volume of items ( $V_{tot}$ ) can be used, respectively. Known data include: total distance of the optimal order picking route ( $d_{tot}$ ), maximal mass ( $m_{max}$ ) and volume ( $V_{max}$ ) one worker can take, total number of available workers, maximal distance

one worker should travel ( $d_{max}$ ), etc. Total order picking route distance  $d_{tot}$  is a solution of TSP, and it represents the distance one worker must travel to complete whole order. This distance does not need to be optimal, since for the sake of the algorithm only good sub-optimal solution is needed. Other data can vary from one warehouse to another, and they can be acquired through conversation with warehouse personnel or from database. Finally, the number of smaller orders is calculated as (1).

$$k = \left\lceil \max \left\{ c_d \cdot \frac{d_{tot}}{d_{max}}, c_m \cdot \frac{m_{tot}}{m_{max}}, c_v \cdot \frac{V_{tot}}{V_{max}} \right\} \right\rceil \quad (1)$$

Parameters  $c_d$ ,  $c_m$  and  $c_v$  are correction coefficients, and have non-negative values. For example, if  $c_m = 0$ , then masses are not important in calculation of  $k$ . This is useful if masses of items are not available. It might be good to set coefficient at 1.1 to 1.3, to gain some flexibility in route calculations.

### C. Order splitting

When the optimal number of smaller order is decided, it is necessary to solve two problems: distribution of items in smaller orders and route optimization for each of them.

Mathematically, the problem can be described as follows. Let  $n$  be the number of different items locations denoted by  $a_1, a_2, \dots, a_n$ . As stated, the calculated number of smaller orders is denoted by  $k$ . The distance between locations  $a_i$  and  $a_j$  is given by  $d_{i,j}$ , for all  $i, j = \overline{1, n}$ . Aggregated mass and volume for items on location  $a_i$  are denoted by  $a_i^m$  and  $a_i^v$ , for all  $i = \overline{1, n}$ .

The algorithm distributes the items in  $k$  by pairs disjoint arrays  $(a_{i,1}, a_{i,2}, \dots, a_{i,l_i})$ , where  $l_i$  denotes the number of locations in  $i$ -th route,  $i = \overline{1, k}$ . Item  $a_{i,j+1}$  is collected just after the item  $a_{i,j}$ , for all  $i = \overline{1, k}$  and  $j = \overline{1, l_i - 1}$ . The starting and ending locations are same for each smaller order because workers have to take the order at the starting location and merge the orders at the end.

Before the process of optimization begins the objective function has to be determined. In practice, the goal is to optimize routes in such a way that the workload of each route is balanced, but at the same time the overall length is as low as possible. A large number of orders can not be optimized in a way that the mass, volume and route length are simultaneously balanced. Therefore, the proposed objective function that is going to be minimized is stated as:

$$D + \max_{i=\overline{1,k}} (c_m \cdot m_i + c_v \cdot V_i + c_d \cdot d_i), \quad (2)$$

where:

- $D$  is the total length of all routes,
- $m_i$  is the total mass for the route  $i$ ,
- $V_i$  is the total volume for the route  $i$ ,
- $d_i$  is the total length of route  $i$ .

Constants  $c_m$ ,  $c_v$  and  $c_d$  are set depending on the needs of the warehouse and the desired criteria. When logistic data are not available, parameter  $c_m$  or  $c_v$  should be set to 0.

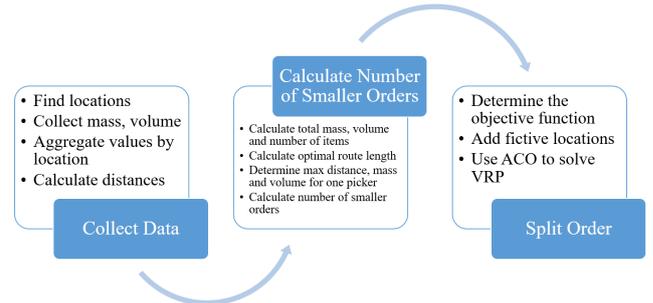


Figure 1. Proposed algorithm

The complete process of items distribution and route optimization is equivalent to the process of solving VRP. The equivalence of customers locations and items locations are established, as is the number of routes with the number of vehicles available. The only difference is the objective function where the goal is to find the balanced routes.

In this paper, the VRP is solved by solving the TSP. The VRP is transformed to TSP by adding starting and ending locations for each route as customers to TSP. Each distance between two added locations is set to  $+\infty$ , so we can guarantee that there will be no optimal route where the added customers are adjacent. The TSP is solved using the Ant Colony Optimization (ACO) [15]. The complete process is shown in Figure 1.

## IV. RESULTS

This section provides a comparison between the proposed algorithm and two approaches that are commonly used in practice for large orders.

1) *One worker*: In many warehouses, the orders are not split despite their size. In case of splitting, the total distance can not be shorter than in the case when one worker is picking the complete order. Meanwhile, orders can have large mass and volume that are greater than the worker capacity. Workers are forced to pause the picking process and unload the picked items on the final location before continuing with the order picking process. In those cases, the traveled distance can be shorter in orders that are split. The main advantage of this approach is the fact that it is not necessary to hire new workers to merge split orders. On the other hand, in many warehouses, some workers are responsible for checking the collected orders and the merging process can be addressed to them.

2) *Divide by sectors*: In practice, the common way of solving the order splitting problem is to divide the warehouse into sectors. The sectors are statically determined, usually by articles type or brand. Then, each sector has a worker in charge and the route is optimized only for that sector. This approach has several drawbacks. The main problem with this approach are orders where majority of locations and items are in one sector. The employee who collects the order in mentioned sector can be overburdened while other employees are spared and their routes are not profitable as they collect

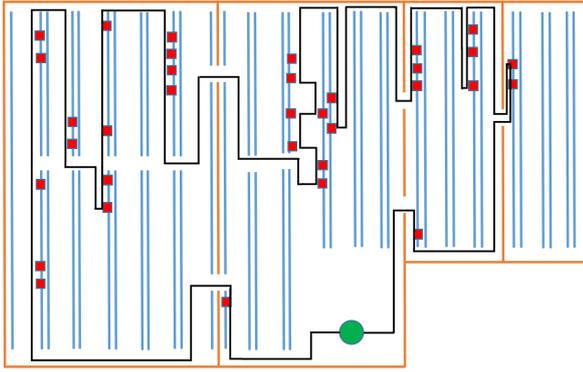


Figure 2. The optimal route for one worker and locations of items

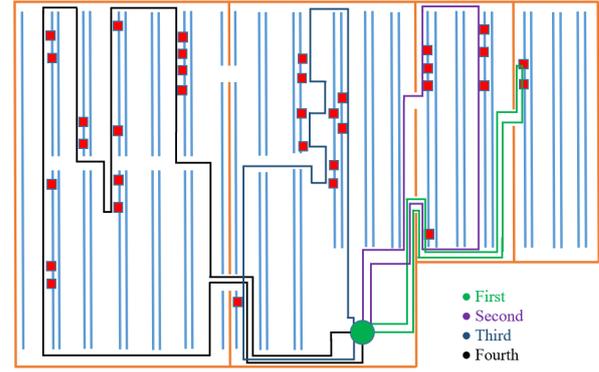


Figure 4. Order divided by sectors

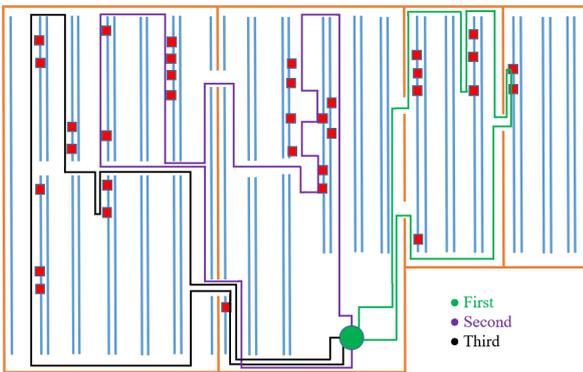


Figure 3. Order with overloads

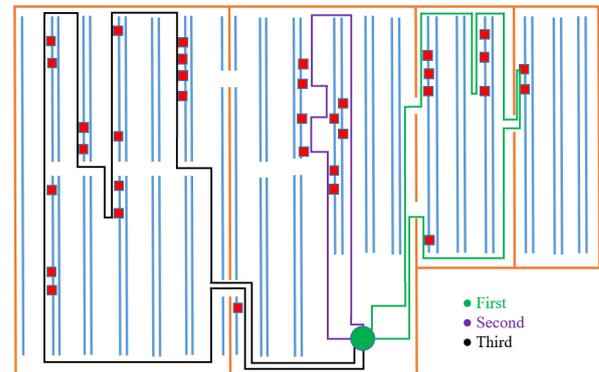


Figure 5. Order given by the proposed algorithm

a small number of items. This approach of order splitting presents the entire warehouse as several smaller warehouses. Workers are better acquainted with their environment and product positions, which is one of the biggest advantages. The sectors are most often created according to the type of goods such as food, animal food, cosmetics, frozen goods etc.

Recognized advantages of the proposed algorithm will be shown on the example of one real-world order. All 34 locations and complete order picking route are shown in Figure 2.

The optimal route length for one worker to complete the order is  $d_{tot} = 1130$  meters. It is important to state that the route was longer than the predetermined optimal route length per worker  $d_{max}$ . It is also observed that the worker reached maximum capacity during the process twice, paused picking and went to unloading location to unload. Due to this action, the route length is increased by 390 meters. Taking this in account, the route with three smaller routes is shown in Figure 3. All constants were determined by the warehouse manager.

When the observed order was split on routes based on the sectors as shown in Figure 4, routes were not balanced. Each room of the presented warehouse was one sector. The total length was 1880 meters. The longest route was 2.75 times longer than the shortest one and the mass was 450% larger.

In the Figure 5 the order splitting by proposed algorithm is given. The total route length was 1430 meters. The longest

route was 2.5 times longer than the shortest one, but the mass was 14% larger. As we can see, the proposed routes are well balanced and the algorithm can solve the drawbacks of earlier approaches. The problems with pausing the order picking process and large distances walked by one worker are solved, as is the problem of unbalanced orders in sectors. At the same time, the total length is not significantly increased.

In addition to the order from the example, the algorithm was tested in one of the largest warehouses in Bosnia and Herzegovina on 1,000 large orders (twenty or more picking locations where one location can contain many items). When warehouse started operating, the orders were not split. Later, the algorithm for splitting to sectors and the proposed algorithm for order splitting were implemented. The same orders were tested using all three mentioned approaches. All 1,000 orders are randomly selected larger orders where logistic data are fully available for easier comparison.

The average number of locations is 34.37 and the average mass is 236.51 kilograms. After solving the TSP and calculating the route length for the TSP, 729 orders exceeded the predetermined limit for route length for one worker. At the same time, 545 orders exceeded mass and volume limit. Detailed results are shown in Table I.

The main observed advantage of this algorithm is the dynamical sector creation and simultaneous reduction worker

Table I  
COMPARISON OF ALGORITHMS FOR 1,000 REAL-WORLD ORDERS IN VARIOUS WAREHOUSE LAYOUTS.

Average	No split	No split (logistic)	Sector	Algorithm	Algorithm (logistic)
Number of smaller routes	1	2.92	4	2.57	2.83
Total distance ( $m$ )	1240	1780	2118	1330	1592
Longest route distance ( $m$ )	1240	1212	1190	540	582
Maximum route mass ( $kg$ )	236.51	126.39	168.22	125	118.23
Maximum route volume ( $m^3$ )	1.72	1.12	1.32	1.12	1.22
Number of times route length exceeded optimal length $d_{max}$	728	542	280	0	0
Walking time to complete order (parallel picking) ( $min$ )	13.52	21.79	12.98	5.89	6.32

load when necessary. This combines the good features of the practice approaches, but at the same time solves their biggest drawbacks. Described algorithm can significantly increase the delivery time if smaller orders are picked at the same time. As stated in [6], most of the orders consist out of 10 or less different items. This means that good positioning of those items can transform the warehouse into smaller one for many orders. In the combination with the proposed algorithm, the maximum distance constant will be rarely exceeded.

The proposed algorithm can be used in a combination with an algorithm for dividing into sectors, where for each smaller order the main sector is determined. The main sector can be determined as the sector where most locations from that order are placed or as the sector where the largest part of the picking route is located. In this way, a primary sector can be assigned to each worker. Then, every order is assigned to a worker with the same primary sector. Each worker can pick some items from locations outside the sector, but the main route will be in the primary sector. This way, the route length is not increasing, but each worker is picking orders in one part of the warehouse, which is emphasized as an advantage of dividing orders by sectors.

## V. CONCLUSION AND FUTURE WORK

In this paper, the motive for order splitting is described, and the complete process of optimal order splitting is presented. The first step was evaluation whether the given order should be split or not, the second was calculating the optimal number of smaller orders it needs to be split into and the final step was assigning items to those smaller orders and optimizing the route for each of the orders. This is implemented as part of the smart WMS in some of the largest warehouses in Bosnia and Herzegovina. The results are satisfying, as well as feedback from warehouse management and workers.

Future research will focus on improving the algorithm with different optimization method. Alongside that, future work will be focused on upgrading mentioned Warehouse Management System with additional features. First of them will be order batching - order picking of many smaller orders or parts of larger orders during one visit, which could be useful for warehouses with 48 hours delivery time, since it cannot be used in real-time.

## REFERENCES

- [1] D. M. H. Chiang, C. P. Lin and M. C. Chen, "The adaptive approach for storage assignment by mining data of warehouse management system for distribution centres," *Enterprise Information Systems* 5.2, 2011. <https://doi.org/10.1080/17517575.2010.537784>
- [2] R. de Koster, T. Le-Duc, and K. J. Roodbergen, "Design and control of warehouse order picking: A literature review," *Eur. J. Oper. Res.*, 2007. <https://doi.org/10.1016/j.ejor.2006.07.009>
- [3] E. Zunic, S. Delalic, K. Hodzic, A. Besirevic, and H. Hindija, "Smart Warehouse Management System Concept with Implementation," in *2018 14th Symposium on Neural Networks and Applications, NEUREL 2018*, 2018. <https://doi.org/10.1109/NEUREL.2018.8587004>
- [4] E. Zunic, A. Besirevic, R. Skrobo, H. Hasic, K. Hodzic, and A. Djedovic, "Design of optimization system for warehouse order picking in real environment," in *ICAT 2017 - 26th International Conference on Information, Communication and Automation Technologies, Proceedings*, 2017. <https://doi.org/10.1109/ICAT.2017.8171630>
- [5] E. Zunic, A. Besirevic, S. Delalic, K. Hodzic, and H. Hasic, "A generic approach for order picking optimization process in different warehouse layouts," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 - Proceedings*, 2018. <https://doi.org/10.23919/MIPRO.2018.8400183>
- [6] E. Zunic, H. Hasic, K. Hodzic, S. Delalic, and A. Besirevic, "Predictive analysis based approach for optimal warehouse product positioning," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 - Proceedings*, 2018. <https://doi.org/10.23919/MIPRO.2018.8400174>
- [7] E. Zunic, K. Hodzic, H. Hasic, R. Skrobo, A. Besirevic, and D. Donko, "Application of advanced analysis and predictive algorithm for warehouse picking zone capacity and content prediction," in *ICAT 2017 - 26th International Conference on Information, Communication and Automation Technologies, Proceedings*, 2017. <https://doi.org/10.1109/ICAT.2017.8171629>
- [8] M. N. Kritikos and G. Ioannou, "The balanced cargo vehicle routing problem with time windows," *Int. J. Prod. Econ.*, 2010. <https://doi.org/10.1016/j.ijpe.2009.07.006>
- [9] I. H. Osman, "Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem," *Annals of operations research* 41.4, 1993. <https://doi.org/10.1007/BF02023004>
- [10] M. Gendreau, A. Hertz, and G. Laporte, "A tabu search heuristic for the vehicle routing problem." *Management science* 40.10, 1994. <https://doi.org/10.1287/mnsc.40.10.1276>
- [11] S. Acharya, "Vehicle Routing and Scheduling Problems with time window constraints-Optimization Based Models," *Int Jr. of Mathematical Sciences Applications* 3.1, 2013.
- [12] M. Desrochers, J. Desrosiers and M. Solomon, "A new optimization algorithm for the vehicle routing problem with time windows," *Operations research* 40.2, 1992. <https://doi.org/10.1287/opre.40.2.342>
- [13] J. E. Bell and P. R. McMullen, "Ant colony optimization techniques for the vehicle routing problem," *Adv. Eng. Informatics*, 2004. <https://doi.org/10.1016/j.aei.2004.07.001>
- [14] M. Yu, "Enhancing Warehouse Performance by Efficient Order Picking," 2008. <https://doi.org/10.1287/opre.37.3.404>
- [15] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem," *BioSystems*, 1997. [https://doi.org/10.1016/S0303-2647\(97\)01708-5](https://doi.org/10.1016/S0303-2647(97)01708-5)