A More Concise and Efficient Formulation of Order Picker Routing in a Rectangular Single-Block Warehouse

Yuqi Liu[1], Haihui Shen[1,2], Jun Xia[1,2]
[1]Sino-US Global Logistics Institute, Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai 200030, China
[2]Data-Driven Management Decision Making Lab, Shanghai Jiao Tong University, Shanghai 200030, China
(yuqi_liu@sjtu.edu.cn; shenhaihui@sjtu.edu.cn; lgtxiaj@sjtu.edu.cn)

Abstract - **Order picker routing in a rectangular single-block (conventional) warehouse is a classical and fundamental logistics problem. Exact algorithm with linear computational complexity exists, and it has also been frequently extended to non-conventional warehouses. This paper proposes a new and more concise formulation of the order picker routing in the conventional warehouse. It is easier to present and understand, based on which the algorithm can be implemented in a more concise way and the computation is more efficient. Viewed as an improvement of existing methods for the order picker routing problem in conventional warehouses, the new formulation and corresponding algorithm have potential to be adopted in non-conventional warehouses.**

Keywords - **Concise formulation, conventional warehouse, order picker routing**

I. INTRODUCTION

Order picking has often been considered as one of the most labor- and time-consuming internal logistics processes, and the order picker's travel time in the warehouse may account for the majority of the total order picking time [1, 2]. As a result, the order picker routing problem has received abundant research in the past; see [3] for a systematic review. In general, the order picker routing problem is a variant of the classical NP hard travelling salesman problem (TSP), but it usually can be solved more efficiently due to the special structure of the aisles in a warehouse. The seminal works that optimally solve the order picker routing problem date back to [4]. They consider a rectangular warehouse with a single block, as illustrated in Fig. 1 (more details later). Moreover, they also assume that: (a) the picking aisles is narrow (i.e., the order picker can pick items from both sides of the aisle without additional travel); (b) the warehouse uses low-level storage racks (i.e., the items can be picked directly from the racks without requiring vertical travels); (c) and the order picking list is static (i.e., what to be picked in a single tour is determined before routing and fixed during the routing). For such a *conventional* warehouse, Ratliff and Rosenthal [4] proposes an exact algorithm whose computational complexity is linear in the number of picking aisles. This algorithm is referred to as RR in the following.

This work is supported by the National Natural Science Foundation of China [Grants 72031006, 72031007, and 72331006].

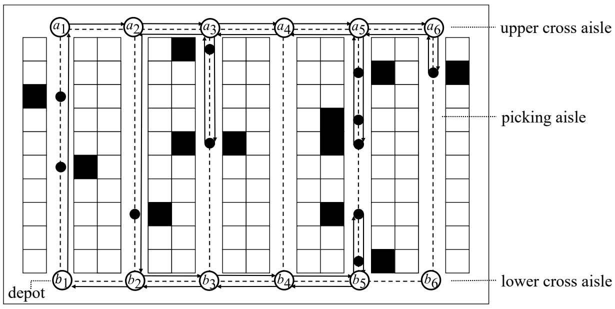

Fig. 1. A rectangular single-block warehouse with 6 parallel picking aisles.

Years have passed, nowadays many warehouses still follow the conventional structure and setup due to the investment cost and operational convenience. So, RR is still a useful algorithm for many practitioners. On the other hand, RR has also been frequently extended to warehouses that are different to the conventional one. For example, [5] extend RR to a warehouse where the order picker can deposit the retrieved items at the respective front end of each picking aisle without returning to the depot; [6] extend RR to consider the precedence constraints in an order picking tour; [7] extend RR to consider the turn penalties; [8] extend RR to account for the dynamic order picking (i.e., the picking list can be updated when it is being completed). Hence, any improvement on RR itself will not only benefit directly the order picking in conventional warehouses, but also provide potential to further improve the order picking in many non-conventional warehouses.

This paper proposes a new formulation of the order picker routing in the conventional warehouse, which is more concise than that in [4]. The value of such new concise formulation is twofold. First, it is easier to present and understand, due to some properties of conventional warehouse and the newly defined state and transition. Second, based on the new concise formulation, the number of required computations to find the optimal order picking route is further reduced. Adopting our new formulation when facing an order picker routing problem in a conventional warehouse, one can implement the algorithm in a more concise way, and the computation is more efficient. More importantly, since RR is a basis of many other algorithms for warehouses more flexible than the conventional one as mentioned before, our new formula-

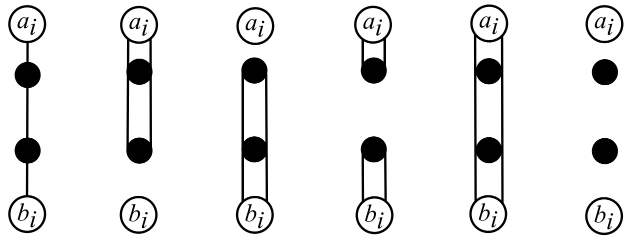

Fig. 2. Possible arc configurations for a picking aisle in an optimal route [4].

tion, which can be viewed as an improvement of RR's formulation, also has potential to be adopted there to improve those algorithms' efficiency.

The remainder of the paper is organized as follows. Section II describes the problem and defines some notations formally. In Section III, the new and concise formulation is established based on two key observations, followed by the corresponding algorithm and its computation complexity analysis. Numerical experiments are conducted in Section IV to demonstrate the higher computation efficiency of the new formulation and verify the derived computation complexity. Section V concludes the paper with an outlook on the future research.

II. PROBLEM

We consider the conventional warehouse as described in the introduction, which is illustrated in Fig. 1. Specifically, the vertical dash lines denote the picking aisles and the horizontal dash lines denote cross aisles, the solid squares denote racks where items have to be picked, and the corresponding dots on the picking aisles denote the *required picking positions*. The warehouse is rectangular literally, which means all the picking aisles are parallel with same length. The depot is both the starting and ending point in an order picking route, which is located at one picking aisle (usually at one of the crossover nodes formed by the picking aisles and cross aisles, e.g., the bottom of the left-most picking aisle). The depot is viewed as one (special) required picking position. Then, a feasible order picking route is a closed tour that starts from the depot, ends at the depot, and contains all required picking positions. One example is illustrated as lines which arrows in Fig. 1. The goal of the order picker routing problem is to find such a route with minimal length (the direction does not matter).

Formally, for the conventional warehouse with n picking aisles (indexed from left to right), define a_i and b_i, $i \in \{1,\dots,n\}$, as the ith crossover nodes on the upper and lower cross aisles, respectively. Define the degree triplet of the ith upper or lower crossover node as (l_v^i, m_v^i, r_v^i), $v \in \{a,b\}$, where $v = a$ means the upper crossover node and $v = b$ means the lower crossover node, l_v^i, m_v^i, and r_v^i denote the left, middle and right degrees of the node respectively (i.e., the number of paths in an order picking route that connects to the node from the left, middle and right, respectively). Clearly, under a feasible order picking route, $l_v^i + m_v^i + r_v^i$ must be even (excluding 0) or 0 for $i \in \{1,\dots,n\}$ and $v \in \{a,b\}$. According to [4], for the ith picking aisle where $i \in \{1,\dots,n\}$, there are totally six possible arc configurations for a picking aisle in an optimal order picking route, as shown in Fig. 2.

III. FORMULATION AND ALGORITHM

[4] propose a formulation based on the idea of dynamic programming, and develop the RR algorithm accordingly. The RR algorithm is an efficient exact algorithm whose computational complexity is linear in the number of picking aisles. We find that the formulation can be further simplified, which can be presented and understood more easily and computed faster. Before introducing the new formulation, we first state and prove two important observations that help to simply the original problem.

Theorem 1. For the conventional warehouse considered in this paper, configuration (v) in Fig. 2 is impossible in an optimal order picking route.

Proof. Suppose the ith picking aisle, $i \in \{1,\dots,n\}$, has configuration (v) in a feasible order picking route.

- Case 1: The route is still connected if the path in the ith picking aisle is deleted. One can replace the configuration (v) with one of the configurations (ii) (iii) (iv) in the ith picking aisle such that the route is still feasible but with smaller length.
- Case 2: The route becomes unconnected if the path in the ith picking aisle is deleted. One of the configurations (ii) (iii) (iv) must exist in one picking aisle that is adjacent to the ith picking aisle, say, the $(i+1)$th picking aisle. Then, by assigning configuration (i) for both the ith and $(i+1)$th picking aisles and adjusting the configuration for the cross aisles between them, one can always obtain a feasible route with smaller length.

Combing the above two cases finishes the proof. ■

Theorem 2. For the conventional warehouse considered in this paper, the picking aisles without required picking points (i.e., empty picking aisles) will not be passed through in an optimal order picking route.

Proof. Suppose the ith picking aisle is an empty picking aisle and it is passed through in an optimal order picking route. By Theorem 1, the possible configuration in the ith picking aisle is (i), (ii), (iii), or (iv). If the configuration is not (i), by changing it to (vi) one can further reduce the route length. So, it is impossible. In the following we will show that the configuration (i) is also impossible, which contradicts the assumption in the beginning and finishes the proof.

If the configuration of the ith picking aisle is (i), there are two cases to be considered.

- Case 1: There is no path on one side of the ith picking aisle. Without loss of generality, suppose $l_a^i = l_b^i = 0$. In this case, by assigning configuration (i) for the $(i+1)$th picking aisle and deleting paths in the cross aisles between the ith and $(i+1)$th picking aisles, one can always further reduce the route length while maintaining its feasibility.

- Case 2: There are paths on both sides of the ith picking aisle. Without loss of generality, suppose l_a^i is even, r_a^i is odd, and r_b^i is odd (while l_b^i can be either even or 0).
  - Consider the subcase where $l_b^i = 0$. Successively check the crossover nodes a_{i-1}, a_{i-2}, …, until find one, say a_j, whose m_a^j is not 0. The configuration of the jth picking aisle will be either (i) or (ii). If it is (i), then delete the path in the ith picking aisle, change the configuration of the jth picking aisle to (v), delete one path between a_j and a_i, and add one path connecting b_j and b_i. As a result, the route will still be feasible and the length remains unchanged. By the same arguments in the proof of Theorem 1, we can then further reduce the route length. If it is (ii), then delete the path in the ith picking aisle, change the configuration of the jth picking aisle to (i), delete one path between a_j and a_i, and add one path connecting b_j and b_i. As a result, the route will still be feasible and the length is further reduced.
  - Consider the subcase where l_b^i is even. Successively check the crossover nodes a_{i-1}, a_{i-2}, …, until find one, say a_{j_1}, whose $m_a^{j_1}$ is not 0, and successively check the crossover nodes b_{i-1}, b_{i-2}, …, until find one, say b_{j_2}, whose $m_b^{j_2}$ is not 0. Let $j = \max\{j_1, j_2\}$. Then, delete the path in the ith picking aisle, change the configuration of the jth picking aisle to (i), and delete one path between a_j and a_i and one path between b_j and b_i. As a result, the route will still be feasible and the length is further reduced.

To sum up, for any case, one can further reduce the route length while maintaining its feasibility. So, the configuration of the ith picking aisle is impossible to be (i). ■

A. Formulation

According to Theorem 2, before searching for the optimal order picking route, we can first delete the empty picking aisles. Without loss of generality, in the following we consider a conventional warehouse with n picking aisles (indexed from left to right) where each picking aisle has at least one required picking position. Moreover, by Theorem 1, we only need to consider configurations (i) to (iv) for a picking aisle in the following analysis.

The construction of a feasible order picking route can be formed by the *decisions* at *n stages*, where in stage $k \in \{1,\ldots, n-1\}$, the decision to make is the route in the kth picking aisle and the upper and lower cross aisles adjacent to the right side, while in stage n the decision is only the route in the nth picking aisle. See an illustration about the stage and decision in Fig. 3.

Suppose we are standing at the beginning of stage $k \in \{2,\ldots, n\}$ and the entire route in the left of the kth picking aisle has been fixed. Then, it is easy to see that the optimal decision in stages $k, k+1,\ldots, n$ to ensure that the final route is feasible and its length is minimized, only depends on two things: (a) the arc configuration on the cross aisles adjacent to the left of the kth picking aisle (i.e., aisles $a_{k-1}a_k$ and $b_{k-1}b_k$); (b) and the number of connected sub routes in the left of the kth picking aisle. So we can define these two pieces of information as a *state* in stage $k \in \{2,\ldots, n\}$, which affects the subsequent decisions. It is easy to see that, in order to finally form a closed route, the state in stage $k \in \{2,\ldots, n\}$ has at most five possibilities, which are denoted as s_1 to s_5 and illustrated in Fig. 4. Note that for state s_1, s_2, s_3, there can only be one connected sub route in the left of the kth picking aisle.

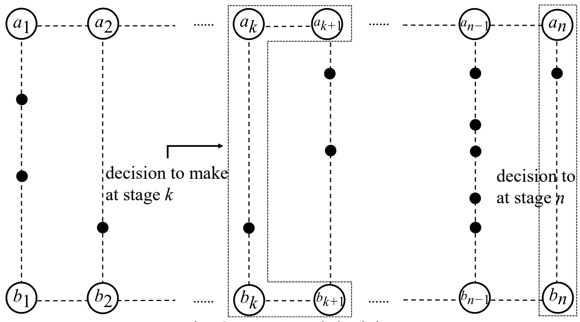

Fig. 3. Stage and decision.

Clearly, in stage $k \in \{2,\ldots, n-1\}$, given the state in this stage, we can make a decision which will determine the state in stage $k+1$. Indeed, from one state in stage $k \in \{2,\ldots, n-1\}$ to one state in stage $k+1$, there may exist multiple choices on the arc configuration for the kth picking aisle or no possible choice, in order to form a feasible route in the end. We present all the scenarios in a transition table (see Table I). In stage 1, there is no state before the decision. By choosing one arc configuration among (i) to (iv), the state in stage 2 will be determined, as shown in Table II. In stage n, the only decision---the route in the nth picking aisle---is uniquely determined by the state in stage n, as shown in Table III.

Compared with that in [4], this formulation is much more concise. This is mainly due to two reasons. The first reason is that with the established Theorems 1 and 2, the possible configurations are reduced. The second reason is that by the newly defined stage, decision and state, the transition is significantly simplified. Such concise is easier to present and understand.

B. Algorithm

Based on the previous defined state and decision in each stage, together with the transition between states, an algorithm that dynamically decides the arc configurations under the new formulation can be easily built and presented.

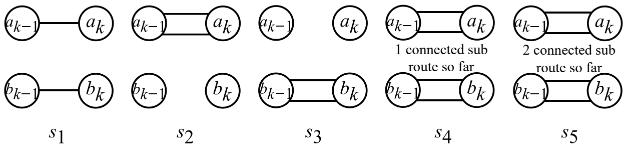

Fig. 4. State in stage $k \in \{2,\ldots,n\}$.

TABLE I

ARC CONFIGURATIONS OF THE kTH PICKING AISLE WHEN STATE TRANSITS FROM STAGE k TO $k+1$, $k \in \{2,\ldots,n-1\}$

State at stage $k+1$ / State at stage k	s_1	s_2	s_3	s_4	s_5
s_1	(i)/(iii)/(iv)	(i)	(i)	(i)	(i)
s_2	(i)	(i)	impossible	(iii)/(iv)	impossible
s_3	(i)	impossible	(i)	(ii)/(iv)	impossible
s_4	(i)	impossible	impossible	(ii)/(iii)/(iv)	impossible
s_5	impossible	(iii)/(iv)	(ii)/(iv)	impossible	(ii)/(iii)/(iv)

TABLE II

ARC CONFIGURATIONS OF THE 1ST PICKING AISLE WHEN STATE TRANSITS FROM STAGE 1 TO 2

State at stage 2 / State at stage 1	s_1	s_2	s_3	s_4	s_5
None	(i)	(ii)	(iii)	impossible	(iv)

TABLE III

ARC CONFIGURATIONS OF THE nTH PICKING AISLE IN STAGE n

State at stage n	Arc configuration of the nth picking aisle
s_1	(i)
s_2	(ii)
s_3	(iii)
s_4	(iv)
s_5	impossible

Algorithm I: Dynamic arc programming for order picker routing in a rectangular single-block warehouse.

Input: Distance matrix and required picking points of the warehouse.
Initialization: Delete the empty picking aisles (n picking aisles left).
for k=1 to n:
  if k==1 **then**
    Initialize a record table as follows, where u_1^1 = (i), u_2^1 = (ii), u_3^1 = (iii), u_4^1 =none, u_5^1 = (iv) according to Table II, and d_1^1 to d_5^1 are computed accordingly.

State in (next) stage 2	s_1	s_2	s_3	s_4	s_5
Arc configuration in previous picking aisle and cross aisles $\mathbf{c}^1$	$\begin{bmatrix} u_1^1 \\ s_1 \end{bmatrix}$	$\begin{bmatrix} u_2^1 \\ s_2 \end{bmatrix}$	$\begin{bmatrix} u_3^1 \\ s_3 \end{bmatrix}$	$\begin{bmatrix} u_4^1 \\ s_4 \end{bmatrix}$	$\begin{bmatrix} u_5^1 \\ s_5 \end{bmatrix}$
Minimal cumulative length to reach the state	d_1^1	d_2^1	d_3^1	d_4^1	d_5^1

  elseif 1<k<n **then**
    For each possible state in next stage k+1, among all possible transitions find the one with smallest cumulative length according to Table I. Update the successive arc configuration and cumulative length.

State in (next) stage k+1	s_1	s_2	s_3	s_4	s_5
Arc configuration in previous picking aisle and cross aisles $\mathbf{c}^k$	$\begin{bmatrix} \mathbf{c}^{k-1}(j_1^k), & u_1^k \\ & s_1 \end{bmatrix}$	$\begin{bmatrix} \mathbf{c}^{k-1}(j_2^k), & u_2^k \\ & s_2 \end{bmatrix}$	$\begin{bmatrix} \mathbf{c}^{k-1}(j_3^k), & u_3^k \\ & s_3 \end{bmatrix}$	$\begin{bmatrix} \mathbf{c}^{k-1}(j_4^k), & u_4^k \\ & s_4 \end{bmatrix}$	$\begin{bmatrix} \mathbf{c}^{k-1}(j_5^k), & u_5^k \\ & s_5 \end{bmatrix}$
Minimal cumulative length to complete the route	d_1^k	d_2^k	d_3^k	d_4^k	d_5^k

  else
    Given state in current stage k (= n), complete and close the route according to Table III. Update the arc configuration and cumulative length.

State in (current) stage n	s_1	s_2	s_3	s_4	s_5
Completed arc configuration	$[\mathbf{c}^{n-1}(j_1^{\mathrm{n}}),u_1^n]$	$[\mathbf{c}^{n-1}(j_2^{\mathrm{n}}),u_2^n]$	$[\mathbf{c}^{n-1}(j_3^{\mathrm{n}}),u_3^n]$	$[\mathbf{c}^{n-1}(j_4^{\mathrm{n}}),u_4^n]$	$[\mathbf{c}^{n-1}(j_5^{\mathrm{n}}),u_5^n]$
Minimal cumulative length to complete the route	d_1^n	d_2^n	d_3^n	d_4^n	d_5^n

  end
end
For the completed arc configuration with the smallest d_j^n, construct the order picking route and return.

Note: 1. Let $d_4^1 = \infty$ as such scenario is impossible.
2. $\mathbf{c}^k$ is the collection of the five elements of that row, and $\mathbf{c}^k(j)$ denote the jth element, j= 1,…, 5, k= 1,…, n.
3. $d_5^n = \infty$ as the route can not be closed in this scenario.

Let u^i, $i \in \{1,\ldots,n\}$ denote the arc configurations of the ith picking aisle, d^k, $k \in \{1,\ldots,n-1\}$ the minimal cumulative length to reach the state in stage k+1, and d^n the minimal cumulative length to complete the route. And subscript is added in u^i, d^k and d^n to denote the possible different scenarios. Then the full algorithm is described in Algorithm I.

C. Computation Complexity

In addition to its conciseness, the new formulation is more efficient in terms of the number of required computations to find the optimal order picking route (i.e., computational complexity), compared to that of RR. As addition and comparison are the most fundamental operations that mainly consume time in both RR and our algorithm, the computational complexity considered in this section refers to the total number of addition and comparison, with assumption that the required picking positions in the warehouse are explicitly given on the diagram of the warehouse.

Consider the stage k as defined in this paper for $1<k<n$. Suppose there are $m\geq 2$ required picking points in the kth picking aisle. First consider the number of additions. It is easy to see that, given the distance matrix, the number of addition in order to compute the length of route in the kth picking aisle for the six configurations shown in Fig. 2 is 0, 1, 1, 3, 1, and 0, respectively; the number of addition in order to compute the length of route in a pair of cross aisles for the five configurations shown in Fig. 4 is 1, 1, 1, 3, and 3, respectively. Inspired by [9], the number of addition required by RR is 49, which consists of 6 addition for the kth picking aisle, 6 addition for the cross aisles, and 37 addition for the state transition (see Tables I and II in [4]). In contrast, the number of additions required by our algorithm is 32, which consists of 5 additions for the kth picking aisle (since configure (v) is eliminated), 6 additions for the cross aisles, and 21 additions for the state transition (since the formulation on state and transition is more concise). Next consider the number of comparisons by assuming the most naive comparison mechanism. Note that in order to determine the configuration (iv), comparison of distances between adjacent required picking points is required. Then, $m-1$ comparison is first required to determine the configuration (iv), and RR subsequently requires 26 comparisons while our algorithm subsequently requires 21 comparisons. To sum up, the total number of computations of RR is $49+m-1+26=74+m$, while that of our algorithm is $32+m-1+21=52+m$. If $m=1$, by similar arguments, it can be found that RR requires 62 computations while our algorithm requires 43 computations. For RR, since the empty picking aisles are not directly deleted, 50 computations are required to handle one empty picking aisle.

So, the overall computational complexity of RR and our algorithm can be characterized as follows. Suppose the originally given warehouse has n picking aisles, and there are m required picking points in each picking aisle. If $m\geq 2$, then RR has computational complexity $(74+m)n+\mathcal{O}(1)$, which is indeed linear in the number of picking aisles as stated in [4], while our algorithm has computational complexity $(52+m)n+\mathcal{O}(1)$. It can be seen that under the new concise formulation, our algorithm requires less computation effort, and the ratio is approximately $(52+m)/(74+m)\in[0.71, 0.72)$ for large n and $2\leq m\leq 5$. If $m=1$, the computational complexity of RR becomes $62n+\mathcal{O}(1)$ and that of our algorithm becomes $43n+\mathcal{O}(1)$. In this case the ratio is still 0.69 for large n. Note that in the originally given warehouse there may exist empty picking aisles. For empty picking aisles RR still needs computations while our algorithm does not. So, for a general warehouse, one may expect a ratio of computational complexity between our algorithm and RR around or smaller than 0.7.

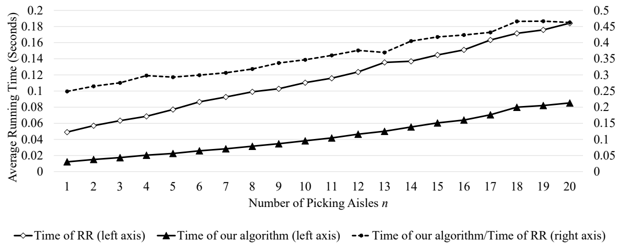

Fig. 5. Time complexity of two algorithms.

IV. NUMERICAL EXPERIMENTS

To demonstrate and verify the theoretical computational complexity derived above, we conduct numerical experiments to investigate the time complexity in practice. Consider conventional warehouses whose total number of picking aisles (i.e., n) varies from 1 to 20. For each n, the number of required picking points in each picking aisle is randomly generated from a discrete probability distribution that has equal probability on $\{0, 1,\ldots,5\}$. To avoid trivial cases, for the 1st and nth picking aisles, the distribution is adjusted to have equal probability on $\{1,\ldots, 5\}$. RR and our algorithm are run, respectively, to search for the optimal order picking route, and their running time is recorded. For each n, we replicate the experiments for 100 times and finally the average running time is calculated. The two algorithms are both plainly implemented in Python 3.9.13 (Windows 11 OS, 3.10 GHz CPU, 16 GB RAM). For every instance, the order picking routes returned by two algorithms have the same length, which should be the minimal length.

Fig. 5 shows the comparison of average running time as n grows from 1 to 20. It is clear that both algorithms' time complexity is linear in the number of picking aisles, but ours has a smaller slope. The time ratio between our algorithm and RR grows as n increases when n is relatively small, because the constant terms in the computational (time) complexity have not be dominated yet (the constant of our algorithm is smaller than that of RR) and their effect gradually fades away. When n gets larger the time ratio tends to approach a constant around 0.47, which is smaller than 0.7 as expected.

V. CONCLUSION

This paper propose a new formulation for the order picker routing problem in a rectangular single-block warehouse. The new formulation is more concise, since some properties of the warehouse are observed and the newly defined state and transition are neater. Under the more concise formulation, the constructed algorithm is more computationally efficient than the existing one (RR), which is shown from both the theoretical computational complexity and the numerical time complexity. As RR has been frequently extended to non-conventional warehouses (see [10] and [11] for example) or other scenarios (see the collaborative robotic picking in [12] for example), the new formulation and corresponding algorithm should also have the potential, which is worth investigating in future research.

ACKNOWLEDGMENT

Haihui Shen is the corresponding author of this paper. The authors thank the anonymous reviewers for their comments and suggestions that enhance the writing of this paper.

REFERENCES

[1] R. De Koster, T. Le-Duc, and K. J. Roodbergen, "Design and control of warehouse order picking: A literature review," European Journal of Operational Research, vol. 182, no. 2, pp. 481–501, Oct. 2007.

[2] Tompkins, J. A., White, J. A., Bozer, Y. A., & Tanchoco, J. M. A.. "Facilities Planning. (4th ed.). John Wiley & Sons", 2010, pp. 386-387.

[3] M. Masae, C. H. Glock, and E. H. Grosse, "Order picker routing in warehouses: A systematic literature review," International Journal of Production Economics, vol. 224, pp. 107564, Jun. 2020.

[4] H. D. Ratliff and A. S. Rosenthal, "Order-Picking in a Rectangular Warehouse: A Solvable Case of the Traveling Salesman Problem," Operations Research, vol. 31, no. 3, pp. 507–521, Jun. 1983.

[5] R. D. Koster and E. V. D. Poort, "Routing orderpickers in a warehouse: A comparison between optimal and heuristic solutions," IIE Transactions, vol. 30, no. 5, pp. 469–480, May 1998.

[6] I. Žulj, C. H. Glock, E. H. Grosse, and M. Schneider, "Picker routing and storage-assignment strategies for precedence-constrained order picking," Computers & Industrial Engineering, vol. 123, pp. 338–347, Sep. 2018.

[7] M. Çelik and H. Süral, "Order picking in a parallel-aisle warehouse with turn penalties," International Journal of Production Research, vol. 54, no. 14, pp. 4340–4355, Jul. 2016.

[8] W. Lu, D. McFarlane, V. Giannikas, and Q. Zhang, "An algorithm for dynamic order-picking in warehouse operations," European Journal of Operational Research, vol. 248, no. 1, pp. 107–122, Jan. 2016.

[9] W. Lu, D. McFarlane, and V. Giannikas, "Interventionist routing algorithm for single-block warehouse: Application and complexity," Technical Report, University of Cambridge, Report No. CUED/E-MANUF/TR.35, pp. 1-16, May 2015.

[10] K. J. Roodbergen and R. de Koster, "Routing order pickers in a warehouse with a middle aisle," European Journal of Operational Research, vol. 133, no. 1, pp. 32–43, Aug. 2001.

[11] Ö. Öztürkoğlu and D. Hoser, "A discrete cross aisle design model for order-picking warehouses," European Journal of Operational Research, vol. 275, no. 2, pp. 411–430, Jun. 2019.

[12] M. Tutam and R. De Koster, "To walk or not to walk? Designing intelligent order picking warehouses with collaborative robots," Transportation Research Part E: Logistics and Transportation Review, vol. 190, p. 103696, Oct. 2024.