A Robust Approach for Improving Computational Efficiency of Order-Picking Problems

Yu-Min Chiang[1], Shih-Hsin Chen[2], and Kuo-Chang Wu[1]

[1] Department of Industrial Engineering and Management, I-Shou University
1, Section 1, Hsueh-Cheng Rd., Ta-Hsu Hsiang,
Kaohsiung County, Taiwan 840, R.O.C
ymchiang@isu.edu.tw, m9320007@stmail.isu.edu.tw
[2] Department of Industrial Engineering and Management, Yuan Ze University
No. 135, Yuan-Tung Rd., Chung-Li, Taoyuan, Taiwan, R.O.C
s939506@mail.yzu.edu.tw

Abstract. The new market forces have affected the operation of distribution centers (DCs) extremely. The DCs demand an increased productivity and a low cost in their daily operation. This paper focuses on the order-picking operation, which is the core operation of DCs, to improve the material-handling process and enhance the competitiveness. The study applied an integrated framework that combined two algorithms, branch-and-bound and tabu search, to the order-picking problem and developed a neighborhood selection criterion based on constraint propagation and the prune-and-search algorithm to improve computational efficiency. The proposed method forbids bad node when searching a solution by branch-and-bound algorithm or tabu search. Besides, in order to make the algorithm more robust and reasonable, the fuzzy modeling was employed. The empirical result shows the approach decreases the computational effort effectively on the both algorithms. The study may be of interest for the researchers in the optimization of DCs' operations and the DCs' managers.

1 Introduction

Distribution centers (DCs) play an important role in modern supply chain environment. It coordinates the manufacturers and customers in the supply chain. However, the material-handling process takes a large proportion of cost, which damaging the survival of DCs [1]. Because the competition is keen in the logistics industry, the study focuses on the order picking operation, which is the core operation of DCs, to improve the material-handling process and enhance their competitiveness.

There are several researches concerning the order-picking problem of DCs [1-5]. Generally, the problems in these models include order-picking within single aisle, weight constraints to zone picking area, number of stock location, maximum load of a picking vehicle, order type, picking strategy, and whole planning of picking area. Previous researchers may use dynamic programming, zero-one integer programming, genetic algorithm, tabu search, and simulated annealing to solve the order-picking problem.

As for the method to improve the computational efficiency, Rana used the weight constraint of picking vehicle to zone picking area [1], thus the problem size could be

R. Khosla et al. (Eds.): KES 2005, LNAI 3682, pp. 1007–1014, 2005.

reduced. Goetschalckx and Ratliff proposed no skip rule to exclude impossible route to increase the efficiency [2]. Although these works can improve the computational efficiency of DCs, it is still not enough for obtaining order picking information in a short time. Therefore, the study developed a framework for order-picking problem which coordinates exact algorithm and heuristics and furthermore proposes a novel approach, which is called neighborhood selection criterion, to improve the computational efficiency. The algorithm avoids algorithms, such as branch-and-bound algorithm or tabu search [6] employed in the research, to select bad nodes before branching or moving a point. Consequently, the approach is useful to reduce the computational effort. Besides, in order to make the algorithm more robust and reasonable, the fuzzy inference system was employed.

The organization of the paper is as followed. Section 2 is the model of the order-picking problem. The methodology is presented in section 3, which is mainly introduces the neighborhood selection criterion the research proposed and the fuzzy inference system. Section 4 is the empirical result and discussion. Section 5 draws the conclusion and guides the future research.

2 Model of Order-Picking Problem

2.1 Assumption of Physical Environment

The research supposes that the width of the aisle should be taken into consideration, and each stock location has its own coordinates, including x, y and z. The I/O or original point O is at (0, 0). The shape of the warehouse is shown as in Fig. 1. There are totally 10 x 10 racks in 2-dimensional space and every rack has three levels. The width of the aisle is 2 meters, and the length of each location is 3, width is 1 and height is 1 meter respectively. As far as the capacity of picking vehicle is concerned, its weight limit is set to 60 kilograms. Besides, the order type is batch picking which collects all orders during a specific time period.

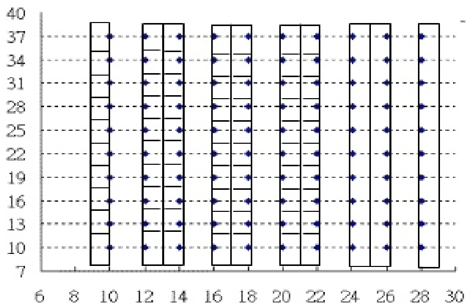

Fig. 1. The shape of the warehouse and the coordinates of each stock location

2.2 Mathematical Model of Order-Picking Problem

The distance between stock locations is a rectilinear distance, not an Euclidian one.

Suppose D_{ij} means the distance from location $i(x_i, y_i, z_i)$ to the location $j(x_j, y_j, z_j)$. A_i and A_j denote the aisle number of the two individual locations respectively. Thus, the rectilinear distance D_{ij} is:

$$D_{ij} = |x_i - x_j| + |y_i - y_j| + (z_i + z_j) \qquad if\ A_i = A_j \text{ , otherwise}$$

$$D_{ij} = |x_i - x_j| + min\{|2Y - y_i - y_j|, |y_i + y_j|\} + (z_i + z_j) \tag{1}$$

Where Y is the total length of an aisle.

The mathematical model of the order-picking is as following:

$$\text{minimize} \quad Z = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} D_{ij} X_{ijk} \tag{2}$$

Subject to

$$\sum_{j=1}^{n} \sum_{k=1}^{n} X_{ijk} = 1,\ i = 0,1,2,\ldots.., \text{n} \tag{3}$$

$$\sum_{i=1}^{n} \sum_{k=1}^{n} X_{ijk} = 1,\ j = 0,1,2,\ldots.., \text{n} \tag{4}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} X_{ijk} = 1,\ k = 0,1,2,\ldots.., \text{n} \tag{5}$$

$$\sum_{i=1, i \neq j}^{n} X_{ijk} = \sum_{r=1, r \neq j}^{n} X_{jr(k+1)} \tag{6}$$

$$\sum_{j}^{n} X_{0j0} + \sum_{j}^{n} X_{(n+1)j0} = 1 \tag{7}$$

Where X_{ijk} means the picking sequence is from location i to location j of the sequence k and its value is either 0 or 1.

3 Methodology

The integrated framework of the research adopts the branch-and-bound algorithm and tabu search. The presupposition is to use cluster algorithm to group items that can be picked by one vehicle [7]. The cluster will contain large size and small size items to be picked, among them the small one is solved by branch-and-bound algorithm and the large one is solved by tabu search. In order to improve computational efficiency, the research proposed a neighborhood selection criterion based on constraint propagation and the prune-and-search algorithm. The proposed method forbids bad node when searching a solution by branch-and-bound algorithm or tabu search. Besides, in order to make the algorithm more robust and reasonable, the fuzzy modeling was employed.

3.1 Neighborhood Selection Criterion

If the branch-and-bound algorithm is able to prune some data points in advance, it is definitely speed up the computation because it no longer has to enumerate impossible points. As for tabu search, it has to move all possible points by local search method. However, if we also set a constraint for the selection of movement, it won't calculate these moves and saves computational time. Consequently, the study attempts to define a neighborhood selection criterion based on constraint propagation and prune-and-search algorithm.

The research lets 65% points be allowed to move and others 35% will be pruned. These 65% values play the role as critical value for later use. The procedures of the neighborhood selection criterion for the branch-and-bound and tabu search are as following:

Step1: Calculate the symmetric distance matrix for each location, including the I/O or the starting point.

Step2: Use the prune-and-search algorithm to find out the distance at the middle position of each location to others in $O(n)$ time-complexity.

Step3: When expanding the possible nodes of branch-and-bound or doing the swap movement of tabu search, it is going to apply the criterion to validate the branching or movement. If it is branch-and-bound algorithm, it goes to step3.1; if the swap movement is backward swap, go to step 3.2; otherwise, go to step 3.3.

- Step3.1: Suppose the current node is i which connects to the point $i+1$. The method checks whether the node $i+1$ locates at the adjacent area to the node i. If yes, execute the branch; otherwise, prune the node we want to branch. Go to step 4.
- Step3.2: Suppose the new position of the moved point is at position i. Check whether the following point, $i+1$, is as the adjacent area to the moved point at position i. If yes, approve the movement; otherwise, discard the movement. Go to step 4.
- Step3.3: Suppose the new position of the moved point is at position i. Check whether the previous point, $i–1$, is as the adjacent area to the moved point at position i. If yes, approve the movement; otherwise, discard the movement. Go to step 4.

Step4: If there are still have nodes needed to branch or move, go to step3. Otherwise, exit the neighborhood selection criterion.

3.2 Fuzzy Modeling Approach

The rule of the neighborhood selection criterion is that the distance between the two nodes is no more than the critical value. However, it may not always robust to sets specific value as critical value. Therefore, the research proposed a fuzzy modeling approach of the distance definition of whether the point is far away. There are inputs, rules, and output in the fuzzy inference system. Figure 2 shows the relationship between the three components.

Input:

The inputs defined here are the rectilinear distance between two nodes and the distance variance. The membership function of inputs is in the pi-shape. Besides, there are three terms, named "Close", "Median", and "FarAway", in the input variable "Distance", and two terms, "Low" and "High", in the input variable "Variance" respectively. There are four parameters of a term to set the pi-shape membership function. Table 1 shows the parameters of the terms of input variable "Distance".

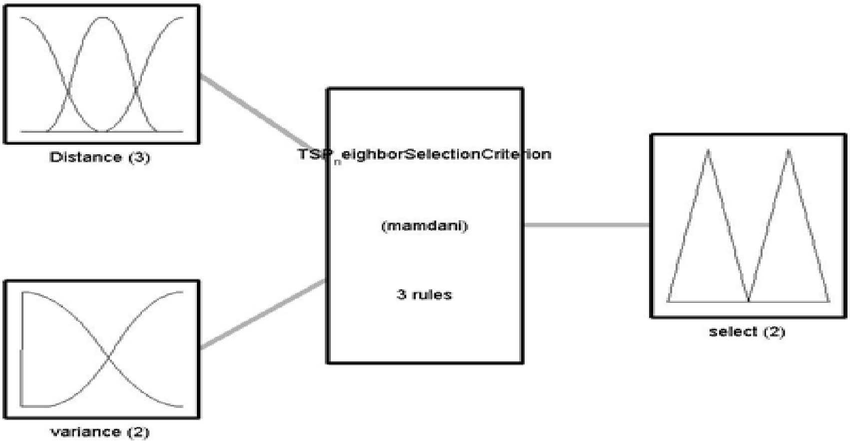

Fig. 2. The fuzzy modeling system

Table 1. The parameters of terms of input variable "Distance"

Term	Parameter of [a, b ,c ,d] of each term
Close	[lower_bound, lower_bound, lower_bound, middle_point];
Median	[middle_point/2, middle_point, middle_point , middle_point*1.5]
FarAway	[middle_point, middle_point, middle_point, upper_bound]

Rules:

The most important component of the fuzzy inference system is the fuzzy rules. It determines interaction between the input variables and the output variable. Three rules are defined in the research: (1) If the distance between *i* and *j* is close, then select it. (2) If the distance between *i* and *j* is far away, then don't select it. (3) If the distance between *i* and *j* is median and the variance is high, then select it. The implementation of the fuzzy model is by the mamdani method and sugeno method.

Output:

The output variable is to determine whether we should select the node. The boundary is between 0 ~ 1. Its attribute includes *selected* or *not selected.* Suppose there are two node *i* and *j*; if the value of output variable is less than 0.5, it is belong to the "not selection" term and $crticalValue_{ij} = 0$ which means the connection between *i* and *j* is forbidden. Otherwise, it accepts the connection and $crticalValue_{ij} = 1$. The membership function of the output variable is in triangular shape.

4 Empirical Result and Discussion

To verify the approach, the research conducts experiments to show the efficiency improving. The programs were written in Java and run on PIII 1.4 G (IBM X220 Dual CPU). Because the branch-and-bound algorithm is the kernel algorithm of ILOG JSolver, the experiment applied the software component to assist the research. Besides, the study also employees a tabu search framework, which is named *OpenTS* [8]. Furthermore, We utilized the fuzzy toolbox of Matlab and use Matlab to call the Java programs of branch-and-bound algorithm and tabu search we wrote rather than implemented them by Matlab language.

The order items are 10 which are randomly generated. Table 2 shows the coordinates of these items. The coordinate of I/O is at (0, 0). The order picker enters from the I/O into the warehouse. After picking the 10 items, the picker goes back to I/O. Although there were only 10 data points, if user wants to evaluate all possible combinations, there are 10! (3628800) possible solutions. The following sections present the implementation performance of branch-and-bound algorithm and tabu search respectively. The two sections show the effect on the computational effort and the solution quality. Then, the research explains the result in the discussion section.

Table 2. Location of the 10 order items

Number	1	2	3	4	5	6	7	8	9	10
X	26	26	14	28	14	22	20	24	10	26
Y	19	25	22	22	28	13	31	25	13	34
Z	1	0	2	0	1	0	0	2	2	1

4.1 The Performance of Branch-and-Bound Algorithm

The original number of nodes of branch-and-bound algorithm was 967670. The objective value of the picking distance is 222 meter. It needs 10.9 seconds to finish all the iteration. The picking sequence is {3, 5, 7, 1, 4, 2, 10, 8, 6, 9}.

Then, the experiment goes to evaluate the performance of the branch-and-bound algorithm when applying the neighborhood selection criterion. The number of choice points becomes 283817 and the computational time is 8.76 seconds. Most important of all, the objective value is still the same. By the above experimental result, we can find out the data nodes needed to be examined that was only 29% compared to original node size and the computational time was decreased 20% in the case. The reason why the saving time is inconsistent with the reduced nodes is that the method requires extra time when it adds constraints to check the neighborhood.

4.2 The Performance of Tabu Search

The section demonstrates the application of the neighborhood selection criterion into tabu search. If the tabu search doesn't apply the approach, it should move all possible moves. The original tabu moves was 45 here and it needed 0.047 seconds in average. However, when the tabu search employees the method, the possible moves of each iteration reduced to 36.72 in average and the computational time was 0.015 seconds. The corresponding improved rate is 17.8% and 32%, respectively. The neighborhood selection criterion approach is able to eliminate unnecessary moves so that increase the computation efficiency

4.3 The Performance of the Fuzzy Approach on Tabu Search

Based on previous experiment result, the work continues the experiment which integrated the fuzzy modeling. The procedures of the examination is the same as Section 4.2. The time-saved is recorded by current cputime() in Matlab. After the implementation of the fuzzy inference system which is mandani type, the objective value is 222 which is still optimal. The number of nodes is 32.68317 in average. Besides, the computational time is 0.2030.

The research found the number of nodes of the fuzzy approach was less than previous one. However, the computational time was ten times higher than before. The reason might be caused by the poor performance of the Matlab operation.

5 Conclusion and Suggestions

The research proposed the neighborhood selection criterion to prune unnecessary nodes. Besides, the fuzzy inference system is also applied in the searching procedure. The empirical result proved the approach is useful on reducing the time complexity and improving the computational efficiency for both the exact algorithm and heuristic. The idea of the algorithm is based on the distance between two locations. When the distance is too far away, the approach rejects the branching or moving action when applying branch-and-bound algorithm or tabu search respectively. In the fuzzy approach, it was proved that it also reduced the number of moves of tabu search. Although the performance is not as good as we only use the tabu search in Java, we may revise the Matlab program into Java ones so that it can improve the performance. The research may be of interest for the researchers in the optimization of DCs' operations and the DCs' managers.

There are some suggestions to make the research be better in the future. First, there are only 10 test order data in the research, the later research can enlarge the problem size and check the solution quality. Second, although we use the fuzzy inference system to assist the criterion, there are some algorithms can be applied. For example, the ANFIS (Adaptive neural network of fuzzy inference system) that adopts the sugeno fuzzy inference system can replace the current FIS model.

References

1. Rana, K., Order-Picking in Narrow-Aisle Warehouse, International Journal of Physical Distribution & Logistics Management, 20(2) (1991) 9-15
2. Liu, C.M., Clustering technique for stock location and order-picking in a distribution center, Computers & Ops Res, 26 (1999) 989-1002

3. Wang, L.H., An integrated storage assignment and order picking sequencing model for distribution centers, master thesis, National Taiwan University of Science and Technology, Taiwan (1998)
4. Chen, L.I., Planning and Evaluation for Manual Picking Operations in Distribution Centers, master thesis, Institute of Industrial Engineering and Management, Yuan-Ze University (2001)
5. Chiang, Y. M. and S. H. Chen, Solving Order Picking Problem by Cluster Analysis and Tabu Search, 2002 National Conference of Management on Technology, HsinChu, Taiwan (2003)
6. Glover, F., Tabu Search II, ORSA Journal on Computing, 2(1) (1990) 4-32
7. Chen, S. H., master thesis, I-Shou University, Taiwan, (2004)
8. OpenTS: http://www-124.ibm.com/developerworks/oss/coin/OpenTS/index.html