American Journal of Operations Research, 2015, 5, 493-502
Published Online November 2015 in SciRes. http://www.scirp.org/journal/ajor
http://dx.doi.org/10.4236/ajor.2015.56038

Mathematical Model and Algorithm for the Task Allocation Problem of Robots in the Smart Warehouse

Zhenping Li, Wenyu Li

School of Information, Beijing Wuzi University, Beijing, China
Email: lizhenping66@163.com, leeyu0513@163.com

Received 10 September 2015; accepted 6 November 2015; published 9 November 2015

Copyright © 2015 by authors and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).
http://creativecommons.org/licenses/by/4.0/

Open Access

Abstract

In the smart warehousing system adopting cargo-to-person mode, all the items are stored in the movable shelves. There are some warehouse robots transporting the shelves to the working platforms for completing order picking or items replenishment tasks. When the number of robots is insufficient, the task allocation problem of robots is an important issue in designing the warehousing system. In this paper, the task allocation problem of insufficient warehouse robots (TAPIR) is investigated. Firstly, the TAPIR problem is decomposed into three sub-problems: task grouping problem, task scheduling problem and task balanced allocation problem. Then three sub-problems are respectively formulated into integer programming models, and the corresponding heuristic algorithms for solving three sub-problems are designed. Finally, the simulation and analysis are done on the real data of online bookstore. Simulation results show that the mathematical models and algorithms of this paper can provide a theoretical basis for solving the TAPIR problem.

Keywords

Smart Warehouse, Warehouse Robot, Task Allocation Problem, Mathematical Model, Heuristic Algorithm

1. Introduction

In the smart warehousing system adopting cargo-to-person mode, all the items are stored in the movable shelves neatly arranged in the warehouse, and picking workers stand in front of stationary picking platforms. There are some isomorphic warehouse robots running along the marked line to complete order picking or items replenishment tasks [1]. When starting order picking or items replenishment tasks, the computer system controls one

How to cite this paper: Li, Z.P. and Li, W.Y. (2015) Mathematical Model and Algorithm for the Task Allocation Problem of Robots in the Smart Warehouse. *American Journal of Operations Research*, **5**, 493-502.
http://dx.doi.org/10.4236/ajor.2015.56038

or more warehouse robots to the sites of assigned movable shelves; the warehouse robots lift up their assigned shelves and transport them to the corresponding picking platforms. After the picking workers complete the order picking or items replenishment tasks, the warehouse robots transport the movable shelves back to their own original locations. In the warehousing management system, complex transportation works are done by the warehouse robots instead of human; the working efficiency is improved in some ways [2].

Since different items of one order might be laid in different shelves. When picking an order, the robots usually need to transport more than one shelf to the picking platforms. In another words, the warehouse robots need to complete multiple tasks for picking one order. In order to improve the picking efficiency, it needs the multi-robot cooperation to complete multiple tasks. The task allocation problem of warehouse robots greatly affects the working efficiency of smart warehousing system.

Some researchers investigated the tasks allocation problem of abundant robots (TAPAR) where the number of robots was larger than the number of tasks to be completed in picking one order. Guo adopts the market auction method to solve the task allocation problem of abundant mobile robots (TAPAR) in a smart logistic centre [3]. Li uses the genetic algorithm and ant colony algorithm to study the TAPAR problem [4]. Christopher *et al.* propose the resource allocation problem of multi-vehicle system in the smart warehouse by using Alphabet Soup method [5]. And Li designs the heuristic algorithm for solving the TAPAR problem [6].

In practical, the number of robots is often smaller than the number of tasks to be completed in picking one order, which means the robots are insufficient. By now, little researches focused on the task allocation problem of insufficient mobile robots (TAPIR).

In this paper, the TAPIR problem in smart warehousing system based on cargo-to-person mode is investigated. We partition the task allocation problem of insufficient robots into three sub-problems: task grouping problem, task scheduling problem and task balanced allocation problem. The mathematical models of three sub-problems are built, and the corresponding heuristic algorithms are designed to solve three sub-problems. Finally, the simulation and analysis are done on the real data of online bookstore *S*.

2. Problem Description and Mathematical Models

2.1. Problem Description

In a smart warehousing system based on cargo-to-person mode, there are m available warehouse robots, k movable shelves, and n picking platforms. Each shelf and picking platform is arranged in a fixed site. Each shelf stores several types of items. The unit-distance cost of a robot walking loaded with a shelf is c_1, the unit-distance cost of a robot walking unloaded is c_2, and the fixed cost of every warehouse robot running one time is r. Let d_{ij} represents the distance of a robot running from shelf s_i to shelf s_j, and u_j represents the distance of a robot transporting shelf s_j to its corresponding picking platform.

The operation cost of a warehouse robot completing a given task is composed of three parts: the related cost, the self-cost and the fixed cost. Related cost refers to the cost that the warehouse robot needs to move unloaded from current position to the position of assigned task. Self-cost refers to the cost that the warehouse robot completing a task. The fix cost refers to the cost running a warehouse robot one time. Suppose the current position of robot F_l is in the position of shelf s_i, if we assign robot F_l to finish task s_j, then the operation cost of robot F_l to finish task s_j can be expressed as follows:

$$c_{lj} = d_{ij}c_2 + u_j c_1 + r$$

where d_{ij} is the distance of robot F_l running from current position s_i to the position of assigned task s_j, u_j is the double distance from position of task s_j to picking platform, r is the fixed cost for controlling robot F_l completing one task.

Given an order to be picked in the smart warehouse, there are p tasks to be completed for picking this order ($m < p$). Assume that the set of tasks for picking this order is $S = \{s_1, s_2, \cdots, s_p\}$.

Suppose there are e orders to be picked in a period of time which are listed in a set: $O = \{O_1, O_2, \cdots, O_e\}$, and the orders are picked one by one, in order to minimize the total operation cost, how to assign all tasks for each warehouse robot and give its working schedule?

The task allocation problem of insufficient warehouse robots (TAPIR) can be decomposed into three sub-problems: task grouping problem, task scheduling problem, and task balanced allocation problem. Firstly, all tasks of picking one order are partitioned into several groups, where the number of groups equals to the number of robots. Secondly, the scheduling scheme of each group tasks is given according to every robot. Finally, assign each warehouse robot to complete one group of tasks.

In the following subsections, we will construct the mathematical models for these three sub-problems.

2.2. Mathematical Model for the Task Grouping Problem

For the task allocation problem of insufficient warehouse robots (TAPIR), since the number of robots is smaller than the number of tasks for picking an order, so some warehouse robots need to complete more than one tasks when picking one order. Before assign the tasks to robots, we firstly partition all tasks into several groups, where the number of group equals to the number of robots.

In order to reduce the total operation cost and ensure the task allocation of warehouse robots balanced, we define the similarity r_{ij} between task s_i and task s_j. The similarity r_{ij} can be calculated by formula (1), where d_{ij} is the distance for a robot to move from the site of task s_i to the site of task s_j, $d_{\max}$ is the maximum distance between any pair of tasks ($d_{\max} = \max\left\{d_{ij}\right\}$).

$$r_{ij} = \frac{d_{\max} - d_{ij}}{d_{\max}} \tag{1}$$

It is easy to verify that $0 \le r_{ij} \le 1$. If we assign two tasks with larger similarity value to one robot, the related cost will be lower. On the contrary, if we assign two tasks with smaller similarity value to one robot, the related cost will be higher. Based on this idea, we can partition the tasks into several groups according to the similarity value. To ensure the task allocation balanced among all robots, we restrict the number of tasks in each group is no more than a given value D (for example $D = \left\lfloor \frac{p}{m} \right\rfloor + 1$).

Define the decision variables:

$$x_{iv} = \begin{cases} 1, & \text{task } s_i \text{ is assigned in the } v\text{th group;} \\ 0, & \text{otherwise } i = 1, 2, \cdots, p; v = 1, 2, \cdots, m. \end{cases}$$

The mathematical model of task grouping problem for picking one order can be formulated as an integer programming model.

$$\max z = \sum_{v=1}^{m} \left(\frac{\sum_{i=1}^{p} \sum_{j=1}^{p} r_{ij} x_{iv} x_{jv}}{\left(\sum_{i=1}^{p} x_{iv} \right)^2} \right) \tag{2}$$

$$s.t. \begin{cases} \sum_{v=1}^{m} x_{iv} = 1, & i = 12, \cdots, p & (3) \\ \sum_{i=1}^{p} x_{iv} \ge 1, & v = 1, 2, \cdots, m & (4) \\ \sum_{i=1}^{p} x_{iv} \le D, & v = 1, 2, \cdots, m & (5) \\ x_{iv} = 0, 1, & i = 1, 2, \cdots, p; v = 1, 2, \cdots, m & (6) \end{cases}$$

Objective function (2) maximizes sum of the average similarity of tasks assigned to the same group; Constraints (3) ensure that each task is just assigned into one group; Constraints (4) and Constraints (5) ensure that each group contains at least one task and no more than D tasks; Constraints (6) indicate that the variables are binary.

2.3. Mathematical Model for the Task Scheduling Problem

Because the self-cost of transporting a task to its corresponding picking platform is a constant no matter which warehouse robot completes this task, the fix cost of each robot is also a constant no matter which task it completes, so the total operation cost of warehouse robots depends mainly on the total related cost that the warehouse robots completing all of the tasks.

If a warehouse robot located in s_0 is assigned to complete a group of tasks $\{s_1, s_2, \cdots, s_q\}$, the total related cost depends on the scheduling order of the tasks. For example, if the scheduling order of the tasks is $s_1 \to s_2 \to \cdots \to s_q$, then the total related cost is $\left(d_{0,1} + d_{1,2} + \cdots + d_{q-1,q}\right) \cdot c_2$. When the scheduling order of tasks changed, the total related cost will change.

Given a warehouse robot and a group of tasks, the task scheduling problem is to find the optimal scheduling order of the tasks, so that the total related cost for the robot completing all tasks is minimal.

Since the related cost depends on the distance between two successive tasks, so the task scheduling problem is similar to Hamiltonian path problem, where the robot should start its routing from initial position s_0, and visit every site of the task in $\{s_1, s_2, \cdots, s_q\}$ one time. After completing all tasks, the robot will stay in the site of the last task, which means that it needs not return to its initial position s_0.

Suppose that warehouse robot F_l (locating in site s_0) is assigned to complete the tasks $\{s_1, s_2, \cdots, s_q\}$ of group k. Suppose that s_{q+1} is a dummy end location, which corresponding to the last location of tasks $\{s_1, s_2, \cdots, s_q\}$. Define $d_{i,q+1} = 0\left(i = 0,1,\cdots,q\right)$.

To find the optimal scheduling order of tasks, we defined the variables as follows:

f_{ij} : the flow that run from task s_i to task s_j, $i = 0,1,\cdots,q;\ j = 1,2,\cdots,q+1$

$$x_{ij} = \begin{cases} 1, & \text{after warehouse robot } F_l \text{ finishing task } s_i \text{, it directly begin task } s_j;\ i = 0,1,2,\cdots,q;\ j = 1,2,\cdots,q,q+1 \\ 0, & \text{Otherwise.} \end{cases}$$

We define that $x_{ij} = \begin{cases} 1, & f_{ij} > 0, \\ 0, & f_{ij} = 0. \end{cases}$

The task scheduling problem can be formulated into the following integer programming model.

$$\min w_{lk} = c_2 \sum_{i=0}^{q} \sum_{j=1}^{q+1} d_{ij} x_{ij} \tag{7}$$

$$s.t \begin{cases} \sum_{j=1}^{q+1} x_{ij} = 1, & i = 0,1,2,\cdots,q & (8) \\ \sum_{i=0}^{q} x_{ij} = 1, & j = 1,2,\cdots,q+1 & (9) \\ \sum_{k=0}^{q} f_{ki} - \sum_{j=0}^{q+1} f_{ij} = 1, & i = 1,2,\cdots,q & (10) \\ \sum_{j=1}^{q} f_{0j} = q, & & (11) \\ x_{ij} \le f_{ij} \le q \cdot x_{ij} & i = 0,1,\cdots,q;\ j = 1,2,\cdots,q+1 & (12) \\ x_{ij} \in \{0,1\} & i = 0,1,\cdots,q;\ j = 1,2,\cdots,q+1 & (13) \end{cases}$$

Objective function (7) minimizes the total related cost that the warehouse robot F_l completes all tasks in group k; Constraints (8) (9) ensure that each task must be completed by robot F_l; Constraint (10) denote that when a task is completed, the total flow run into it will be absorbed 1 by the task. Constraint (11) ensures the sum of flows come out from location s_0 equals q. Constraint (12) represents the relationship between x_{ij} and f_{ij}; Constraint (13) indicates that the variables x_{ij} are binary.

2.4. Mathematical Model for the Task Balanced Allocation Problem

After calculating the related cost w_{lk} of warehouse robot $F_l\left(l = 1,2,\cdots,m\right)$ to complete the tasks in group $k\left(k = 1,2,\cdots,m\right)$, we obtained a related cost matrix $W = \left(w_{ij}\right)_{m\times m}$. Since we have partitioned the tasks into m groups, each robot needs to complete one and only one group of tasks. The tasks balanced allocation problem is to assign each robot completing only one group of tasks so that the total related costs is minimal.

Define the decision variables:

$$y_{lk}=\begin{cases}1, & \text{assign warehouse robot } F_l \text{ to complete the } k\text{th group of tasks;}\\ 0, & \text{otherwise } l=1,2,\cdots,m;\ k=1,2,\cdots,m\end{cases}$$

The mathematical model of tasks balanced allocation problem can be formulated into an integer programming model as follows:

$$\min z=\sum_{l=1}^{m}\sum_{k=1}^{m}w_{lk}y_{lk} \tag{14}$$

$$s.t.\begin{cases}\sum\limits_{l=1}^{m}y_{lk}=1, & k=1,2,\cdots,m & (15)\\ \sum\limits_{k=1}^{m}y_{lk}=1, & l=1,2,\cdots,m & (16)\\ y_{lk}\in(0,1), & l=1,2,\cdots,m,\ k=1,2,\cdots,m & (17)\end{cases}$$

Objective function (14) minimizes the total related cost that warehouse robots complete all groups of tasks; Constraints (15) ensure that each group of tasks can only be assigned to one warehouse robot; Constraints (16) represent that each warehouse robot can only complete one group of tasks; Constraints (17) indicate that the variables are binary.

3. Algorithms for Solving Three Sub-Problems

In Section 2, we have decomposed the TAPIR problem into three sub-problems and formulate them as integer programming models. For small size of sub-problems, we can solve the integer programming model by Lingo software. Since some sub-problem (such as the second sub-problem) is NP-hard, for larger size of the problem, it is not practical to solve them by Lingo software. In this section, we will respectively design heuristic algorithms for solving each sub-problem of Section 2.

3.1. Algorithm for Solving the Task Grouping Problem

Given the number of robots *m* and the position of each robot, the set of tasks $\{s_1,s_2,\cdots,s_p\}$ need to be completed in picking one order. The heuristic algorithm for grouping all tasks into *m* groups can be described as follows:

Initialization: $T=\{s_1,s_2,\cdots,s_p\}$, *group* = 0, where *T* is the set of ungrouped tasks, *group* denotes the number of non-empty groups.

Step 1: Calculate the similarity matrix $R=(r_{ij})_{p\times p}$, where r_{ij} is the similarity between task s_i and task s_j, which can be calculated by Equation (1).

Step 2: Find the smallest value of matrix R, $r_{i_0,j_0}=\min\limits_{i,j}\{r_{ij}\}$, let $G_1=\{s_{i_0}\}$, $G_2=\{s_{j_0}\}$; $group=2$; $T=T\setminus\{s_{i_0},s_{j_0}\}$.

Step 3: If *group* = *m*, go to step 4, else.

For each $s_i\in T$, and $k=1,\cdots,group$, calculate the average similarity between the ungrouped task s_i and all tasks in group *k* as follows

$$AR(i,k)=\frac{\sum_{s_j\in G_k}r_{ij}}{|G_k|}.$$

The sum of average similarity between task s_i and all non-empty groups

$$SAR(i)=\sum_{k=1}^{group}AR(i,k).$$

Find the task with minimum sum of average similarity

$$i_{\min} = \arg\min_{i} SAR(i).$$

Update the number of non-empty group, add the tasks in the new non-empty group, and update the set of ungrouped tasks.

$$group := group + 1;$$
$$G_{group} := \{s_{i_{\min}}\};$$
$$T := T \setminus \{s_{i_{\min}}\};$$

Go to step 3.

Step 4: If $T = \Phi$, go to step 5. Else add the ungrouped tasks to the group, with which it has the maximum average similarity.

For each $s_i \in T$, and $k = 1, \cdots, m$, where $|G_k| < D$.

Calculate the average similarity between s_i and each task in group k.

$$AR(i,k) = \frac{\sum_{s_j \in G_k} r_{ij}}{|G_k|}.$$

Find the group with maximum average similarity value.

$$AR(i^*, k^*) = \arg\max_{i,k} AR(i,k).$$

Update the set T and G_{k^*}

$$T := T \setminus \{s_{i^*}\},\quad G_{k^*} := G_{k^*} \cup \{s_{i^*}\}.$$

Go to Step 4.

Step 5: Output the tasks in each group: $G_1, G_2, \cdots, G_m$.

3.2. Algorithm for Solving the Task Scheduling Problem

To solve the task scheduling problem, we design a heuristic algorithm:

Define the following variables:

$route(l,k)$ represents the scheduling route, whose initial value is empty;

$location$ represents the location of robot F_l, whose intimal position is s_{l0};

w_{lk} represents the total related cost for robot F_l completing tasks in $G_k = \{s_{k_1}, s_{k_2}, \cdots, s_{k_{gk}}\}$;

Input: The locations of m robots and the sites of each task in m groups.

Step 1: For each robot $F_l\ (1 \le l \le m)$ and each task group $G_k\ (1 \le l \le m)$, suppose the location of robot F_l is s_{l0}, the number of tasks in group k is g_k, $G_k = \{s_{k_1}, s_{k_2}, \cdots, s_{k_{gk}}\}$.

Let $TN = \{k_1, k_2, \cdots, k_{g_k}\}$ be the index set of tasks.

$$route(l,k) = \Phi;\quad location = 0;\quad w_{lk} = 0.$$

Calculate the distance matrix $D(l,k) = (d_{ij})_{(g_k+1)\times(g_k+1)}$, where d_{0j} is the distance from s_{l0} to s_{k_j}, d_{ij} is the distance from s_{k_i} to s_{k_j}, $(i, j = 1, 2, \cdots, g_k)$.

Step 2: If $TN = \Phi$, go to step 3.

Else, find the minimal element of matrix $D(l,k)$ in the *location* row and columns of TN,

$j^* = \arg\min_{j \in T}\{d_{location,j}\}$, $route(l,k) := route(l,k) \cup \{j^*\}$, $TN := TN \setminus \{j^*\}$, $location := j^*$, $w_{lk} := w_{lk} + c_2 d_{location,j^*}$, go to step 2.

Step 3: Output the total related cost w_{lk}, the optimal scheduling order of tasks $route(l,k)$, and the final location of robot F_l.

3.3. Algorithm for Solving the Task Balanced Allocation Problem

After scheduling the tasks of each group, we obtain the related cost matrix $W = (w_{lk})$, where w_{lk} represents the related cost of robot F_l completing the tasks of groups k. Then the balance allocation problem is converted into an assignment problem, which can be solved by Hungarian Algorithm [7]. We can also obtain the optimal solution by solving the integer linear programming model directly.

4. Simulations Analysis

In this section, we do simulation on a smart warehouse of online bookstore *S*. At present, online bookstore *S* uses the smart ware housing system based on cargo-to-person mode. In the warehouse, 96 classes of books are stored in 24 movable shelves. Each shelf stores 4 classes of books. There are 3 picking platforms in the smart warehouse. Currently, the orders are picked one by one. **Figure 1** depicts the layout of the warehouse, the square boxes represent 24 shelves and their positions, three rectangular boxes represent three picking platforms; three cycles with arrow represent the warehouse robots. The coordinates of shelves and picking platforms are listed in **Table 1**. The initial positions of three robots are the same as their nearest shelves.

There are 10 orders to be picked. The tasks sets of 10 orders are listed in **Table 2**. The unit cost of a robot walking loaded with a shelf is 3 dollars, the unit cost of a robot walking unloaded is 1.9 dollars, and the fixed cost of controlling each warehouse robot completing a task is 10 dollars. In order to minimize the total cost in the process of picking 10 orders, how to allocate and schedule the tasks to each warehouse robot?

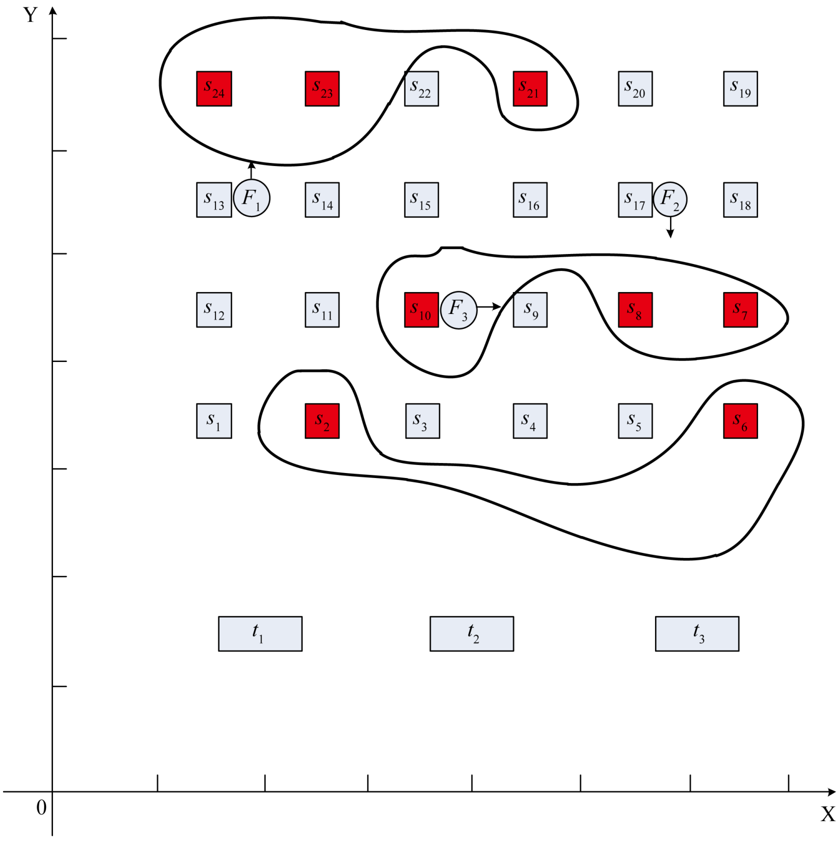

Figure 1. The layout of the warehouse, the square boxes represent shelves, three rectangular boxes represent three picking platforms three cycles with arrow represent the warehouse robots. 8 tasks of the first order depicted in red squares are partitioned into three groups. The tasks in the same group are surrounded by a closed curve (Coordinate Unit: 2 m).

Table 1. Coordinates of each shelf and each picking platform.

Shelf	s_1	s_2	s_3	s_4
Coordinate	(3, 7)	(5, 7)	(7, 7)	(9, 7)
Shelf	s_5	s_6	s_7	s_8
Coordinate	(11, 7)	(13, 7)	(13, 9)	(11, 9)
Shelf	s_9	s_{10}	s_{11}	s_{12}
Coordinate	(9, 9)	(7, 9)	(5, 9)	(3, 9)
Shelf	s_{13}	s_{14}	s_{15}	s_{16}
Coordinate	(3, 11)	(5, 11)	(7, 11)	(9, 11)
Shelf	s_{17}	s_{18}	s_{19}	s_{20}
Coordinate	(11, 11)	(13, 11)	(13, 13)	(11, 13)
Shelf	s_{21}	s_{22}	s_{23}	s_{24}
Coordinate	(9, 13)	(7, 13)	(5, 13)	(3, 13)
Picking platform	t_1	t_2	t_3	
Coordinate	(4, 3)	(8, 3)	(12, 3)	

Table 2. The tasks of picking each order.

Order	Tasks of picking the order
1	$\{s_2, s_6, s_7, s_8, s_{10}, s_{21}, s_{23}, s_{24}\}$
2	$\{s_2, s_5, s_{10}, s_{16}\}$
3	$\{s_4, s_8, s_{12}, s_{18}, s_{20}, s_{22}, s_{24}\}$
4	$\{s_3, s_9, s_{14}, s_{18}, s_{19}, s_{22}, s_{24}\}$
5	$\{s_1, s_3, s_5, s_{11}, s_{14}, s_{15}, s_{19}, s_{23}\}$
6	$\{s_2, s_5, s_9, s_{11}, s_{16}, s_{19}, s_{23}\}$
7	$\{s_4, s_7, s_{10}, s_{13}, s_{17}, s_{18}\}$
8	$\{s_2, s_8, s_{14}, s_{22}, s_{23}\}$
9	$\{s_2, s_6, s_{11}, s_{12}, s_{14}, s_{19}, s_{22}, s_{23}\}$
10	$\{s_3, s_6, s_7, s_{12}, s_{17}, s_{21}\}$

Since the orders are picked one by one, for each order to be picked, we can calculate the distance between any pair of tasks, and obtain the similarity matrix by Equation (1).

For the *u*th $(u = 1, 2, \cdots, 10)$ order, we first partition the tasks into three groups by the heuristic algorithm in section 3.1.

Then we obtain the scheduling order for each group tasks and every robot by the heuristic algorithm in section 3.2. Using the Hungarian Algorithm, we get the tasks assignment and scheduling results, and the total operation cost.

After assigning the tasks of the *u*th order to the robots, we update the initial position of each robot, and begin to assign the tasks of the $(u+1)$ th order. Repeat the steps until all tasks of 10 orders are assigned to robots.

The simulation results of the first order is depicted in **Figure 1**, where 8 tasks are partitioned into three groups, the tasks in the same group are surrounded by a closed curve. The optimal tasks assignment and the scheduling routes are $F_1 \to s_{24} \to s_{23} \to s_{21}$, $F_2 \to s_6 \to s_2$, $F_3 \to s_{10} \to s_8 \to s_7$. Which mean that robot F_1 need to complete tasks s_{24}, s_{23}, s_{21} in sequence, then stop at position s_{21}. Robot F_2 need to complete tasks s_6, s_2 in sequence, then stop at position s_2. Robot F_3 need to complete tasks s_{10}, s_8, s_7 in sequence, then stop at position s_7.

The task grouping results of all orders are listed in **Table 3**. The task allocation and optimal scheduling routes of all orders are shown in **Table 4**, the total operation costs of each order are listed in **Table 4**.

According to the task assignment and scheduling routes, and the operation cost in **Table 4**, the total operation cost that the warehouse robots complete all tasks of 10 orders is 4330.2 dollars.

If the number of warehouse robots is larger than the maximum number of tasks for picking one order, the problem becomes TAPAR. The tasks need not be grouped, and the total operation cost for picking these 10 orders will be 4010.4 dollars [2].

From the task allocation results of 10 orders, we can see that the total operation cost in TAPIR is 319.8 dollars more than the total operation cost in TAPAR [2]. However, the number of warehouse robots in the TAPIR is 7 less than that in TAPAR. Since the cost of increasing a warehouse robot is very large, in practice, the number of warehouse robots is often insufficient. If we consider holding cost of robots in the TAPIR, we can find that the warehouse had better hold fewer robots.

Table 3. Task grouping results of 10 orders.

Order	Group 1	Group 2	Group 3	Sum of average similarity
1	$\{s_{21},s_{23},s_{24}\}$	$\{s_7,s_8,s_{10}\}$	$\{s_2,s_6\}$	2.416667
2	$\{s_2\}$	$\{s_5,s_{16}\}$	$\{s_{10}\}$	2.625000
3	$\{s_{20},s_{22},s_{24}\}$	$\{s_{18}\}$	$\{s_4,s_8,s_{12}\}$	2.333333
4	$\{s_{19},s_{22},s_{24}\}$	$\{s_{18}\}$	$\{s_3,s_9,s_{14}\}$	2.333334
5	$\{s_{15},s_{19},s_{23}\}$	$\{s_5,s_{11},s_{14}\}$	$\{s_1,s_3\}$	2.319444
6	$\{s_{16},s_{19},s_{23}\}$	$\{s_{11}\}$	$\{s_2,s_5,s_9\}$	2.428572
7	$\{s_{13},s_{17}\}$	$\{s_4,s_{10}\}$	$\{s_7,s_{18}\}$	2.416666
8	$\{s_{22},s_{23}\}$	$\{s_{14}\}$	$\{s_2,s_8\}$	2.500000
9	$\{s_{19},s_{22},s_{23}\}$	$\{s_{11},s_{12},s_{14}\}$	$\{s_2,s_6\}$	2.333333
10	$\{s_3,s_{12},s_{17}\}$	$\{s_6,s_7\}$	$\{s_{21}\}$	2.472222

Table 4. The task assignment and optimal scheduling routes, and the operation cost of each order.

Order	Task assignment and optimal scheduling routes	Operation cost
1	$F_1 \to s_{24} \to s_{23} \to s_{21}$; $F_2 \to s_6 \to s_2$; $F_3 \to s_{10} \to s_8 \to s_7$	517.2
2	$F_1 \to s_{16} \to s_5$; $F_2 \to s_2$; $F_3 \to s_{10}$	222.6
3	$F_1 \to s_{20} \to s_{22} \to s_{24}$; $F_2 \to s_4 \to s_8 \to s_{12}$; $F_3 \to s_{18}$	508.2
4	$F_1 \to s_{24} \to s_{22} \to s_{19}$; $F_2 \to s_{14} \to s_9 \to s_3$; $F_3 \to s_{18}$	493.6
5	$F_1 \to s_{19} \to s_{23} \to s_{15}$; $F_2 \to s_3 \to s_1$; $F_3 \to s_5 \to s_{11} \to s_{14}$	512.8
6	$F_1 \to s_{16} \to s_{19} \to s_{23}$; $F_2 \to s_2 \to s_5 \to s_9$; $F_3 \to s_{11}$	457
7	$F_1 \to s_{13} \to s_{17}$; $F_2 \to s_7 \to s_{18}$; $F_3 \to s_{10} \to s_4$	381.6
8	$F_1 \to s_{22} \to s_{23}$; $F_2 \to s_8 \to s_2$; $F_3 \to s_{14}$	361.2
9	$F_1 \to s_{23} \to s_{22} \to s_{19}$; $F_2 \to s_2 \to s_6$; $F_3 \to s_{14} \to s_{11} \to s_{12}$	514
10	$F_1 \to s_{21}$; $F_2 \to s_6 \to s_7$; $F_3 \to s_{12} \to s_3 \to s_{17}$	362

5. Conclusions

The task allocation problem of insufficient warehouse robots (TAPIR) in the smart warehouse is an important issue in the smart warehousing system designing. By now, few researches focus on the TAPIR problem. In this paper, we investigate the TAPIR problem, and divide it into three sub-problems: task grouping problem, task scheduling problem and task balanced allocation problem. We formulate each sub-problem into an integer programming model and design the heuristic algorithm for solving each sub-problem. The mathematical models and algorithms can be used in designing the smart warehousing system.

In this paper, we assume that the orders are picked one by one. When the number of orders is large, the orders can be batched firstly [8], and then consider the orders in each batch as one combined order. Using the mathematical models and heuristic algorithms of this paper, we can find the optimal task allocation and scheduling routes for picking each combined order.

In this paper, we have not considered the cost of holding a robot. If we consider this cost, we can analyze the relation between the total cost and the number of warehouse robots and find the optimal number of robots a warehouse should hold. This is the problem we are researching on.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (11131009, F012408), the Funding Project for Academic Human Resources Development in Institutions of Higher Learning Under the Jurisdiction of Beijing Municipality (CIT&TCD20130327), and Major Research Project of Beijing Wuzi University. Funding Project for Technology Key Project of Municipal Education Commission of Beijing (ID: TSJHG201310037036); Funding Project for Beijing key laboratory of intelligent logistics system; Funding Project of Construction of Innovative Teams and Teacher Career Development for Universities and Colleges Under Beijing Municipality (ID: IDHT20130517); Funding Project for Beijing philosophy and social science research base specially commissioned project planning (ID: 13JDJGD013).

References

[1] Li, Z. and Li, W. (2014) Study on Optimization of Storage Bays of Smart Warehouses of Online Bookstores. *Logistics Technology*, **12**, 340-342. (In Chinese)

[2] Zou, S. (2013) The Present and Future of Warehouse Robot. *Logistics Engineering and Management*, **6**, 171-172. (In Chinese)

[3] Guo, Y. (2010) Auction-Based Multi-Agent Task Allocation in Smart Logistic Center. Ph.D. Thesis, Harbin Institute of Technology, Shenzhen. (In Chinese)

[4] Li, G. (2012) Task Allocation of Warehouse Robots Based on Intelligence Optimization Algorithm. Ph.D. Thesis, Harbin Institute of Technology, Shenzhen. (In Chinese)

[5] Hazard, C.J., Wurman, P.R. and D'Andrea, R. (2006) Alphabet Soup: A Tested for Studying Resource Allocation in Multi-Vehicle Systems. *American Association for Artificial Intelligence*.

[6] Li, Z. and Li, W. (2015) Research on the Task Allocation Problem of Warehouse Robots in the Smart Warehouse. *Proceedings of the* 12*th International Symposium on Operations Research and Its Applications*, Luoyang, 21-24 August 2015, 29-33.

[7] Kuhn, H.W. (2005) The Hungarian Method for the Assignment Problem. *Naval Research Logistics*, **1**, 7-21. http://dx.doi.org/10.1002/nav.20053

[8] Li, Z. and Li, W. (2014) Study on Supplier Warehouse Location Combination Problem of Chain Supermarkets. *Logistics Technology*, **9**, 137-139. (In Chinese)