Research on Task Assignment Model and Algorithm of Online Picking System Based on Multi-objective

Jianyun Zhang, Zhuan Wang* and Xiang Xiao

School of Mechanical Engineering, University of Science and Technology Beijing, Beijing 100083, China

*Corresponding author e-mail: wangzhuan@ustb.edu.cn

Abstract. In view of the task assignment problem of the picking station in the online Parts-to-Picker Picking System, a mathematical model was established that targeting the minimum number of total rounds of the bins and the equalization of the picking of each station, constrained by the uniqueness of the order attribution and the maximum amount of orders processed by the workstation. Then, the solution flow and method of multi-objective function problem based on Genetic Algorithm are designed. The algorithm constrains the secondary objective function and defines the rules of the objective function constraint. Finally, an empirical study on the actual Parts-to-Picker System of an e-commerce enterprise is carried out. The results show that the algorithm can effectively reduce the two objective function values and verify the effectiveness of the algorithm.

1. Introduction

With the rapid development of e-commerce and new retail models, the Online Parts-to-Picker Picking System is becoming more and more widely used in various industries, but the lack of scientific picking plans and perfect task allocation methods often make the application effect greatly compromised. In actual use, the design capability is far from being achieved, resulting in waste of resources.

In the Online Parts-to-Picker Picking System, the picking strategy is no longer suitable. To achieve the goal of balancing the workload of each picking station and reducing the frequency of container warehousing (the number of hoppers in the container), it is a new research point to improve the efficiency of picking work—what method and strategy are used to assign the divided wave order to each picking station. At present, there is little research involved in this problem.

At present, the existing research results on this issue have the following points: (1) Assign the order to the workstation according to the variety storage location. WANG Wenrui [1] designed a straight-through sorting station with AS/RS to study the problem of matching the storage location of items in the roadway with the order allocation between the various picking stations. A multi-stage heuristic algorithm is proposed, and the method is consistent with the idea of picking and arranging people, but the new picking station designed to pass through the storage system has a complex structure and is not universal. (2) Assign the order to the workstation with the cache station as the prototype. Wu Yingying [2] proposed an order sorting model, which defines the order coupling factor to represent the number of common containers that can be placed in the picking cache between two orders, and uses this as a model parameter to reduce the container inbound and outbound frequency. The method takes into account both
the order allocation and the order sorting of the picking station, but it also needs to set the storage buffer area for designing the new picking station, which takes up a large space and does not conform to the current mainstream picking station. (3) The order processing problem of a single picking station. Specifically, it refers to the dynamic matching of the outbound container and the order processing in the Parts-to-Picker Picking System, that is, in a single picking station, the orders are processed to meet the matching between the outbound container arrival sequence and the picking order picking sequence. David Füßler [3] studied the problem of synchronous matching of outbound containers and picking orders in the Parts-to-Picker Picking System, and proposed a mixed integer model based on the problem, and solved the order with a two-stage heuristic algorithm. Scientific processing allows for the maximum number of pickers to be reduced under all order fulfillment conditions. Based on the Kiva system, Nils Boysen [4] studied the synchronization matching of shelves and picking orders in the Parts-to-Picker System, and proposed a heuristic algorithm to optimize the order sequence to realize the dynamic matching between the order processing and the outbound shelves. And pointed out that future research should study the overall system, including picking workstations, transport systems and storage systems, rather than just solving each element optimization problem in isolation. [5].

In summary, the current solution to the TAP (Task Assignment Problem) of the picking station is based on the specific type of Parts-to-Picker System and picking station. The design of the picking station is not universal, and the research has not clearly defined the Order Task Assignment problem of the picking station. A more focused research point is the order processing problem of a single picking station. However, there is little research on TAP.

In terms of solving methods, the most representative multi-objective genetic algorithms include multi-objective genetic algorithms PESA2 [6] and NSGA2 [7] [8]. The advantage of PESA2 is that its solution has good convergence and can often be close to the optimal surface, especially in the case of high dimensional problems. However, the disadvantage is that the selection operation can only select one individual at a time, the time is expensive, and the diversity of the solution set is not good. When on the high-dimensional problem, the diversity of the solution set of NSGA2 is not ideal because of the defects of the aggregation process. The model in this paper has two objective functions, but the functions do not always conflict or are not always related, and their degree of correlation is affected by order data and inventory data. Therefore, this paper transforms the Task Assignment model into a single-objective problem, and then uses the single-objective optimization algorithm to solve the model.

This paper takes the TAP of picking station tasks in the online Parts-to-Picker Picking System as the research object. It aims to establish a mathematical model of multi-objective Task Assignment through in-depth research and propose a scientific and effective genetic algorithm to enrich and improve the Parts-to-Picker Picking System.

2. Problem Description
The Online Parts-to-Picker Picking System generally consists of two parts: the storage system and the sorting system, as shown in Fig. 2-1. According to the wave orders, the storage system completes the delivery of the bin by the working equipment (stacker or trolley/hoist) and then transports it to the picking station via the conveyor system. After picking up the required quantity of goods from the bin according to the order information assigned to the workstation, the picking station transports the bin to the next picking station via the transport system for picking until the bin picking task is completed. The bin is then returned to the storage port by the conveyor and stored by the warehouse operation equipment.

The TAP is to determine how to assign a batch of orders to each picking station in the case of known outbound task list and order information, so that the picking efficiency of the whole batch is the highest. For different picking forms, the main difference is the number of orders to be allocated, among them, the collection order can be regarded as an order with a large number of orders. Therefore, this paper combines them when discussing the TAP of the Parts-to-Picker System picking station.
In the TAP, the order information and the outbound task list are known information, and the order allocation result of each picking station is to be optimized. When establishing the mathematical model of the problem, first define the relevant parameters in the known information. Then select appropriate optimization goals according to the actual operation process, and then make rational assumptions for the optimization goals. Finally, the relevant constraints are listed, and the mathematical model of the TAP of the picking station under the Parts-to-Picker System is given.

3. Model Establishment

3.1. Parameter Definition

First, the known information is defined as follows.

(1) Item information

Define the set $I$ to represent all the items to be picked in a batch. For each item $i \in I$ to be picked, I define $Q_i$ to indicate the total quantity of the item.

(2) Order information

Define the set $R$ to represent all the orders to be allocated in a batch, then the number of elements in the set $R$, $|R|$ indicate the number of orders to be allocated for the batch. For each order to be assigned $r \in R$. Define $q_{ir}$ to indicate the ordered quantity of item $i$ in order $r$, then the total ordered quantity of order $r$ is $Q_r = \sum_{i \in I} q_{ir}$; the total number of items of item $i$ is $Q_i = \sum_{r \in R} q_{ir}$.

(3) Outbound task list

Define the set $J$ to represent all the bins in the library list (the storage unit in most Parts-to-Picker Systems is the bin, and the AS/RS is the pallet). For each bin $j \in J$, define a subset $I_j$ of the set $I$ to represent all the items to be picked in the bin $j$, and define $p_{ij}$ to represent the number of items to be shipped out of the item $i$ in the bin $j$, then for each item $i$ needs to satisfy $\sum_{j \in J} p_{ij} = Q_i$.

(4) Picking station

Define the set $S$ to represent all the picking stations in the Parts-to-Picker System, then the number of elements in the set $S$, $|S|$ represents the number of picking stations. For each picking station, $s \in S$ defines $Q_s$ as the upper limit of the order quantity (the number of picking station sorting points or seeding wall sorting points) that the station can process simultaneously.
3.2. Mathematical model

The result of solving the TAP is to give an order allocation plan. So for each order to be allocated \( r \in R \), define the two-dimensional decision variable \( x_{rs} \) to indicate whether the order \( r \) is assigned to the picking stations. If the order \( r \) is assigned to the picking stations, then \( x_{rs} = 1 \), if not, then \( x_{rs} = 0 \), ie:

\[
x_{rs} = \begin{cases} 
1, & \text{the order } r \text{ is assigned to the picking station } s \\
0, & \text{the order } r \text{ is not assigned to the picking station } s
\end{cases}
\]

To facilitate the representation and calculation of the number of tours of each bin, define a two-dimensional process variable \( h_{js} \) to indicate whether bin \( j \) has to pass through the picking stations. The process variable \( h_{js} \) can be obtained from the decision variable \( x_{rs} \) by certain rules (the rules will be detailed later). If the bin \( j \) needs to pass the picking stations, then \( h_{js} = 1 \); otherwise, \( h_{js} = 0 \), ie:

\[
h_{js} = \begin{cases} 
1, & \text{the bin } j \text{ needs to pass the picking station } s \\
0, & \text{the bin } j \text{ needs not to pass the picking station } s
\end{cases}
\]

Based on the above variables and parameters, the mathematical model for establishing the task assignment problem is as follows:

**Objective function:**

\[
\min\left(\sum_{j \in J} \left( \sum_{s \in S} h_{js} \right) \right)
\]

\[
\min\left(\max_{s \in S} Q_r x_{rs} \right)
\]

**Restrictions:**

\[
\sum_{s \in S} x_{rs} = 1, \quad \forall r \in R
\]

\[
\sum_{s \in S} x_{rs} = 1, \quad \forall r \in R
\]

\[
\sum_{r \in R} x_{rs} < O_s, \quad \forall s \in S
\]

\[
x_{rs} \in \{0, 1\}, \quad \forall j \in J
\]

Among them:

The objective function (1) indicates that the number of tours of the bin is the least;

The objective function (2) represents the equalization of the picking amount of each picking station, and the minimum value of the single station picking quantity is selected as the objective function. \( Q_r = \sum_{i \in I} q_{ir} \) is the total ordered quantity of order \( r \); \( \sum_{r \in R} Q_r x_{rs} \) is the picking workload of the picking station \( s \);

The constraint (3) formula indicates that an order is only allowed to be assigned to one workstation;

The constraint (4) formula indicates that the number of orders allocated to each picking station is not allowed to exceed the upper limit of the order quantity that can be processed simultaneously;

The constraint (5) formula specifies the binary and non-negative nature of the value of \( x_{rs} \).

3.3. Variable Relationship

There are three cases of bins to multiple workstations, multiple bins to one workstation, and multiple bins to multiple workstations. Wherein, when one bin is connected to a plurality of workstations and a plurality of bins to one workstation, the variable \( h_{js} \) can be obtained by a logical operation as shown in equation (6) by the variable \( x_{rs} \) to be solved.
\( h_{js} = \bigvee_{i \in I} \left( p_{ji} \bigvee_{r \in R} (q_{ir} x_{rs}) \right) \) \hspace{1cm} (6)

However, there are allocation problems for multiple bins for multiple workstations, so the relationship between \( h_{js} \) and \( x_{rs} \) cannot be obtained by direct arithmetic or logical operations. Therefore, for the above three cases, the \( h_{js} \) value of each bin is obtained uniformly by the following logic rules:

1) Defining the two-dimensional variable \( q_{is} \) indicates the number of items \( i \) to be picked up at the picking station \( s \), then the relationship between \( q_{is} \) and \( x_{rs} \) can be expressed as: \( q_{is} = \sum_{r \in R} q_{ir} x_{rs} \);

2) \( h_{js} = 0, \forall j \in J, \forall s \in S \);

3) For each item \( i \in I_j \) in each bin \( j \in J \):
   1) For each picking station \( s \in S \):
      a) If \( q_{is} \) or \( p_{ij} \) is equal to 0, skip the current workstation;
      b) If both \( q_{is} \) and \( p_{ij} \) are not equal to 0, execute:
         a) If \( q_{is} \geq p_{ij} \), then: \( q_{is} = q_{is} - p_{ij} \); \( p_{ij} = 0 \);
         b) If \( q_{is} < p_{ij} \), then: \( p_{ij} = p_{ij} - q_{is} \); \( q_{is} = 0 \).

The flowchart and pseudo code of the \( h_{js} \) assignment rule are shown in Fig. 3-1. According to the above logic rules, the \( h_{js} \) value of each bin can be obtained. The number of tours of bin \( j \) can be expressed as \( \sum_{s \in S} h_{js} \), then the total number of tours of all bins can be expressed as \( \sum_{j \in J} \left( \sum_{s \in S} h_{js} \right) \).

Therefore, the objective function with the least number of rounds of the bin is shown in equation (1) in the mathematical model.
4. Multi-Objective Genetic Algorithm Design

In this paper, the Genetic Algorithm is used to optimize the proposed task distribution model of the pick-up picking station of the goods-to-person online picking system. The optimization goal is to reduce the number of rounds of the picking box, and balance the picking quantity of each picking station as a constraint condition.

In response to the task assignment model proposed in this paper, the following innovations and adjustments are made on the flow of the general Genetic Algorithm:

1. For the case of two objective functions in the model, the “ITI (Initial Tolerance Index)” parameter is introduced, and the constraint of the objective function (2) is defined.

2. In the individual selection, a new comparison operator is defined in consideration of the constraint of the upper limit of the function value of the objective function (2).

3. When evenly crossing, the number of defined intersections is randomly generated between $R/3$ and $2R/3$, and $R$ is the length of the chromosome.

4.1. Objective Function Constraint

For the objective function (2), the second objective function is constrained by specifying an upper limit of its function value and adding the upper limit to the constraint condition.

4. Multi-Objective Genetic Algorithm Design

In this paper, the Genetic Algorithm is used to optimize the proposed task distribution model of the pick-up picking station of the goods-to-person online picking system. The optimization goal is to reduce the number of rounds of the picking box, and balance the picking quantity of each picking station as a constraint condition.

In response to the task assignment model proposed in this paper, the following innovations and adjustments are made on the flow of the general Genetic Algorithm:

1. For the case of two objective functions in the model, the “ITI (Initial Tolerance Index)” parameter is introduced, and the constraint of the objective function (2) is defined.

2. In the individual selection, a new comparison operator is defined in consideration of the constraint of the upper limit of the function value of the objective function (2).

3. When evenly crossing, the number of defined intersections is randomly generated between $R/3$ and $2R/3$, and $R$ is the length of the chromosome.

4.1. Objective Function Constraint

For the objective function (2), the second objective function is constrained by specifying an upper limit of its function value and adding the upper limit to the constraint condition.
Since the randomly generated solution also performs poorly on the objective function (2), the upper limit of the objective function (2) function value in the constraint process cannot be a fixed value and must be a decreasing variable.

To achieve this reduction in the upper limit, a new parameter is defined: \( \epsilon \). The objective function (2) function value upper limit decrement rule is as follows:

1. Calculate the lower limit \( \text{Optima}_2 \) of the objective function (2) function value, that is, the value of the objective function (2) in the case of complete equalization, \( \text{Optima}_2 = \text{total number of batch picks} / \text{number of workstations}, \) rounded up;
2. Specify the upper limit \( \text{Limit}(0) \) of the objective function (2) function value in the initial population as:

\[
\text{Limit}(0) = \text{Optima}_2 \times (1 + 0.05)^\epsilon
\]

3. The ITI is uniformly reduced to 1 as the number of iterations, that is, the upper limit \( \text{Limit}(g) \) of the objective function (2) function value in any generation \( g \) is:

\[
\text{Limit}(g) = \text{Optima}_2 \times (1 + 0.05)^\epsilon \frac{g^{\epsilon}(\epsilon - 1)}{\text{Gens}}, \text{Gens} \text{ is the total number of iterations.}
\]

### 4.2. Chromosome Coding

The order allocation is to select the appropriate picking station \( s \) for each order, so the chromosomal encoding can be performed using the picking station number \( s \) assigned to each order. So this article uses an order-based integer encoding, the length \( L \) of the chromosome is equal to the number of orders to be allocated in the batch \( |R| \), Each gene position of the chromosome represents an order, and the gene value represents the sorting station to which the order is assigned, and the range of each gene values on the chromosome is Integer on the interval \([1, |S|]\). When \( |R| = 15 \), chromosome encoding and decoding are shown in Fig.4-1.

![Figure 4-1. Chromosome encoding and decoding](image)

The Initial population generation strategy, Individual selection strategy, Cross operation strategy, Variation operation strategy adopted by the Genetic Algorithm in this paper is shown in Table 4-1.

| Table 4-1. The strategy of Genetic Algorithm |
|---------------------------------------------|
| Initial population generation | Randomly generated |
| Individual selection strategy | Tournament selection |
| Cross operation design | Uniform crossover |
| Variation operation design | Mutation probability \( p_m \) |

### 5. Algorithm Verification

The empirical study is based on actual order data for an e-commerce company and a list of outbound tasks. Through the algorithm parameter experiment, the best performing parameter combination is sought, and different batch orders under different storage strategies are optimized to verify the rationality and effectiveness of the algorithm.

The Genetic Algorithm optimization program uses the Spyder integrated development environment, written in Python language, and the experimental environment is the same as the empirical study of the
outbound bin assignment problem. In the process of optimizing the task assignment problem of the goods picking station, the outbound task list and the order information to be allocated are known and stored in the Access database. For the outbound task list information, there are still two cases of no LCL storage and LCL storage, which will be optimized separately.

5.1. Convergence Curve and Evaluation Method of Optimization Effect

(1) Convergence Curve

Through the genetic algorithm designed for the task assignment model, the order of each batch of two batches is optimized without LCL storage and LCL storage. In the crossover rate of 0.9, the mutation rate is taken as 0.2, the population size is taken as 100, the league size in the tournament is taken as 2, and the ITI is taken as 10. In the case of 500 iterations, the convergence curve shown in Fig. 5-1 and Fig. 5-2 is obtained. It can be seen from the convergence curve that the algorithm basically converges after 200 generations.

![Figure 5-1. Convergence curve of the Genetic Algorithm optimization task assignment model (Without LCL storage)](image1)

![Figure 5-2. Convergence curve of the Genetic Algorithm optimization task assignment model (LCL storage)](image2)

(2) Optimization Effect Evaluation Method

However, in the design of the algorithm, the multi-objective problem is converted into a single-objective problem for optimization, but when evaluating the optimization effect, two objective functions should still be considered. The optimal solution of the problem to be solved in this paper is unknown, so the optimization effect is evaluated by calculating the optimization percentage of the two objective
functions with respect to the initial solution. The calculation process of the evaluation indicators is as follows:

1) Let the two objective functions of the optimal individual in the initial population be \( f_1, f_2 \);
2) Calculate the \( hv \) value of the global optimal individual \( i \) after the optimization is completed according to the following formula, \( f_1(i) \) and \( f_2(i) \) are the two objective function values of the individual:
\[
hv(i) = \frac{r_1 - f_1(i)}{r_1} + \frac{r_2 - f_2(i)}{r_2}
\]

It can be seen from the calculation process that the optimal effect of the optimal individual compared with the initial solution is better, and the \( hv \) value is larger. A large \( hv \) value does not necessarily mean that the individual performs better on the objective function, but rather that the optimal solution is more optimized than the initial solution.

5.2. Algorithm Parameter Experiment

The population size, crossover rate, mutation rate and the ITI in the objective function constraint in the genetic algorithm will have an impact on the optimization of the task assignment problem of the Parts-to-Picker System picking station. Therefore, this paper separately sets different parameter values for the four parameters of crossover rate, mutation rate, population size and ITI. Then choose the parameter value with better average performance as the parameter value optimized for the task assignment problem of the goods-to-person system picking station.

For other parameters in the genetic algorithm, the following are set in the experimental phase and the problem optimization phase:

1) The size of the league in the tournament selection method: 2;
2) Number of intersections in a uniform crossover operation: a random integer between chromosome length/3 and chromosome length *2/3;
3) The number of mutation points in the mutation operation: chromosome length/5;
4) Number of algorithm iterations: 300.

For the study data, select batch order with batch size of 300 without LCL storage and LCL storage. The number of picking stations is 10, and the order limit for each picking station can be 40. Calculate the mean value of the objective function of the two batches of optimal individuals without LCL storage and LCL storage as the criterion for optimizing the effect. For any batch, the algorithm runs 3 times and takes the average of 3 run results.

(1) Crossover Rate Experiment

This section uses the crossover rate design as the basis for crossover rate selection. In the experiment, the mutation rate was set to 0.2, the population size was 100, and the ITI was 9. The crossover rate was increased from 0.6 to 1 in steps of 0.1, and the best performing crossover rate was selected from the experimental results. The crossover rate experimental results are shown in Table 5-1.

Table 5-1. Crossover Rate test results

| Cross Rate | 0.6 | 0.7 | 0.8 | 0.9 | 1   |
|------------|-----|-----|-----|-----|-----|
| Optimal solution objective function 1 mean | 187 | 173.5 | 172.5 | 160 | 161 |
| Optimal solution objective function 2 mean | 72.5 | 68.5 | 74.5 | 70  | 70.5 |
| \( hv \) mean | 0.5188 | 0.5179 | 0.4604 | 0.6128 | 0.5942 |

As can be seen from Table 5-1, the algorithm optimization performs best when the crossover rate is 0.9.

(2) Mutation Rate Experiment

This section uses the design variation rate experiment as the basis for the selection of the mutation rate. In the experiment, the crossover rate was set to 0.9, the population size was 100, the ITI was 9, and the mutation rate was increased from 0.04 to 0.2 in steps of 0.04. The best performing mutation rate was
selected from the experimental results. The experimental results of the mutation rate are shown in Table 5-2.

| Mutation Rate | 0.04 | 0.08 | 0.12 | 0.16 | 0.2 |
|---------------|------|------|------|------|-----|
| Optimal solution objective function 1 mean | 180.5 | 178 | 168.5 | 168 | 164.5 |
| Optimal solution objective function 2 mean | 75 | 79 | 71 | 69 | 69.5 |
| $hv$ mean | 0.4756 | 0.4657 | 0.6058 | 0.5415 | 0.6279 |

It can be seen from Table 5-2 that the algorithm optimization performs best when the mutation rate is 0.2.

(3) Population Size Experiment

This section uses the design of population size experiments as the basis for population size selection. In the experiment, the crossover rate was set to 0.9, the mutation rate was 0.2, and the ITI was 9. The population size increased from 50 to 200 in 50 steps. The results of the population size experiment are shown in Table 5-3 and Fig. 5-4.

| Population Size | 50 | 100 | 150 | 200 |
|-----------------|----|-----|-----|-----|
| Optimal solution objective function 1 mean | 186 | 167 | 155.5 | 154.5 |
| Optimal solution objective function 2 mean | 77 | 69.5 | 69.5 | 73.5 |
| $hv$ mean | 0.4382 | 0.6235 | 0.5635 | 0.6039 |
| operation time | 105.7 | 217.2 | 324.3 | 422.8 |

It can be seen from the results in Table 5-3 that algorithm optimization performs best when the population size is 150. At the same time, it can be seen from Table 5-3 and Fig. 5-3 that the larger the population size, the longer the algorithm optimizes the operation and the less likely it is to converge.

(4) ITI experiment

This section uses the design of the ITI experiment as the basis for the ITI selection. In the experiment, the crossover rate was set to 0.9, the mutation rate was 0.2, and the population size was 100. The ITI increased from 7 to 10 in steps of 1. The experimental results of the ITI are shown in Table 5-4.
Table 5-4. ITI experimental results

| ITI  | 7    | 8    | 9    | 10   |
|------|------|------|------|------|
| Optimal solution objective function 1 mean | 174  | 166.5| 167  | 163.5|
| Optimal solution objective function 2 mean | 68   | 70   | 69.5 | 72.5 |
| hυ mean | 0.5538 | 0.5654 | 0.6235 | 0.5261 |

Unlike the previous three experimental parameters, the ITI is a parameter used to constrain the objective function (2). The larger the ITI, the more tolerant the value of the objective function (2) is, the objective function (1) is relatively easier to obtain better values. The experimental results in Table 5-4 also verify this. Then consider the two objective function considerations and choose an ITI of 9.

5.3. Optimization Effect Analysis

In this section, we select the order of batches of 300, no LCL storage and LCL storage, and use the optimal parameter combination obtained by parameter experiment to optimize and solve the problem by means of genetic algorithm, respectively showing LCL storage and LCL Storage optimization results.

(1) No LCL storage

The function optimization effects of the five batch target functions (1) and the objective function (2) without LCL storage are shown in Table 5-5 and Table 5-6 respectively.

Table 5-5. Optimization effect of each batch target function (1) without LCL storage

| Batch number | 1    | 2    | 3    | 4    | 5    | Average |
|--------------|------|------|------|------|------|---------|
| Initial population | 306  | 299  | 275  | 276  | 299  | 293     |
| Optimal solution  | 178  | 197  | 177  | 185  | 190  | 187     |
| Optimization ratio | 42%  | 34%  | 35%  | 33%  | 36%  | 36%     |

Table 5-6. Optimization effect of each batch of objective function (2) without LCL storage

| Batch number | 1    | 2    | 3    | 4    | 5    | Average |
|--------------|------|------|------|------|------|---------|
| Initial population | 98   | 132  | 96   | 94   | 100  | 104     |
| Optimal solution  | 69   | 88   | 68   | 70   | 72   | 73      |
| Optimization ratio | 30%  | 33%  | 29%  | 26%  | 28%  | 29%     |

As can be seen from Table 5-5 and Table 5-6, the two objective functions of each batch without LCL storage are optimized. Among them, the average optimization percentage of the objective function (1) reaches 36%, and the average optimization percentage of the objective function (2) reaches 29%.

(2) LCL storage

The function optimization effects of the five batch target functions (1) and the objective function (2) in the LCL storage are shown in Table 5-7 and Table 5-8 respectively.

Table 5-7. Optimization effect of each batch of objective function (1) in LCL storage

| Batch number | P1    | P2    | P3    | P4    | P5    | Average |
|--------------|-------|-------|-------|-------|-------|---------|
| Initial population | 253  | 247   | 241   | 236   | 252   | 246     |
| Optimal solution  | 148   | 157   | 153   | 162   | 179   | 160     |
| Optimization ratio | 42%  | 36%   | 37%   | 31%   | 29%   | 35%     |

Table 5-8. Optimization effect of each batch of objective function (2) in LCL storage

| Batch number | P1    | P2    | P3    | P4    | P5    | Average |
|--------------|-------|-------|-------|-------|-------|---------|
| Initial population | 98   | 132   | 96    | 94    | 100   | 104     |
| Optimal solution  | 69    | 88    | 68    | 70    | 72    | 73      |
| Optimization ratio | 30%  | 33%   | 29%   | 26%   | 28%   | 29%     |
Table 5-8. Optimization effect of each batch of objective function (2) in LCL storage

| Batch number | P1  | P2  | P3  | P4  | P5  | Average |
|--------------|-----|-----|-----|-----|-----|---------|
| Initial population | 101 | 138 | 88  | 89  | 95  | 106     |
| Optimal solution    | 70  | 91  | 66  | 66  | 70  | 74      |
| Optimization ratio  | 31% | 34% | 25% | 26% | 26% | 29%     |

As can be seen from Table 5-7 and Table 5-8, the two objective functions of each batch under the LCL storage are optimized. The average optimization percentage of the objective function (1) reaches 35%, and the average optimization percentage of the objective function (2) reaches 29%.

6. Conclusion
In this paper, based on the online picking system of parts-to-picker, a task assignment model of online picking system based on double target with the least number of rounds of the tank and the balance of workstation tasks is established. The “ITI (Initial Tolerance Index)” parameter is introduced, and the objective function (2) is constrained. The Genetic Algorithm of multi-objective optimization is designed. The effectiveness of the algorithm is verified by empirical research. The empirical research shows that the algorithm has obvious optimization effects under the two storage strategies of LCL and LCL.

References
[1] Wang W, Wu Y, Wu Y. A multi-stage heuristic algorithm for matching problem in the modified miniload automated storage and retrieval system of e-commerce [J]. Chinese Journal of Mechanical Engineering, 2016, 29 (3): 641 - 648.
[2] Wu Yingying, Meng Xiangxu, Wang Yanyan, et al. Order Sorting Optimization of "Parts to picker" Picking System [J]. Journal of Mechanical Engineering, 2016, 52 (4): 206 - 210.
[3] David Füßler, Nils Boysen, Efficient order processing in an inverse order picking system, Computers & Operations Research, Volume 88, 2017, Pages 150-160, ISSN 0305 - 0548.
[4] Boysen N, Briskorn D, Emde S. Parts-to-picker based order processing in a rack-moving mobile robots environment [J]. European Journal of Operational Research, 2017: S0377221717302758.
[5] Füßler, David, Boysen N. High-performance order processing in picking workstations [J]. EURO Journal on Transportation and Logistics, 2017.
[6] D. W. Corne, N. R. Jerran, J. D. Knowles and M. J. Oates, PESA-II: Region-based selection in evolutionary multi-objective optimization, InProceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001), Morgan Kaufmann Publishers, 2001, pp. 283-290.
[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGAII, IEEE Trans. Evol. Comput, vol.6, Apr. 2002., pages 182 - 197.
[8] Deb K., Agrawal S., Pratap A., & Meyarivan T. A Fast Elitist Non-Dominated-Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II, KanGAL Report No. 200001, 2000.