Multi-task Scheduling of Agvs System Based on Improved NSGA-II Algorithm

Tingge Ren¹,a,* and Yueyang Ren²,b
¹ChongQing University,ChongQing,China
²Ningxi Junior Middle School,HuangYan,China
E-mail: a,*20161302015@cqu.edu.cn, bren6912268110163.com

Abstract. Aiming at the problem that the AGV energy consumption is not considered in the multi-task mode, a multi-task model is established which takes the total energy consumption of AGV and the total time of tasks out of warehouse as the goal, and the task group reconstruction is realized based on the batch-combination strategy. Then, this paper improves the NSGA-II algorithm to solve the model from the following three aspects: population screening mechanism, pheromone-based crossover, and double mutation operate. The experimental simulation results show that the improved NSGA-II algorithm improves the quality of the solutions, and meanwhile improves the stability of the algorithm compared with the original NSGA-II algorithm.

1. Introduction
With the rapid development of "Internet +" technology, people’s consumption concept has gradually shifted from physical shopping to online shopping. The upgrading of industrial structure and consumption structure has promoted the rapid development of e-commerce, express delivery and other industries. At the same time, with the continuous improvement of people’s living standards, the strength of commodity purchase is gradually strengthened, which puts forward higher requirements for the transportation efficiency of the logistics system.

In the early work, most of the multi-task scheduling goals were based on minimizing the outbound time and driving distance [1–3], and the problem of AGV energy consumption was ignored in the model. The higher the energy consumption of AGV, the more times of charging in the operation process, which would cause system interruption and thus reduce the picking efficiency. In recent years, the importance of problems related to AGV energy consumption in the scheduling process has been studied [4–6], and AGV energy consumption has been taken as one of the scheduling objectives [7–9]. However, the influence of AGV start-stop and load on AGV energy consumption has not been considered in the existing models. Therefore, this paper establishes a multi-task model that minimizes the total AGV energy consumption and the total task outbound time, and solves the model through the improved NSGA-II algorithm.

2. Problem descriptions and assumptions
In this paper, a multi-task scheduling model is established based on minimizing the overall AGV power consumption and minimizing the task delivery time.
2.1. Assumptions

- Ignore the volume of the transported cargo, only consider the weight of the cargo, and the weight of a single cargo does not exceed the maximum carrying capacity of the AGV.
- Each AGV travels at the same speed and runs at a constant speed. The speed of the straight section and the turning section are the same, regardless of the influence of the turning section on the AGV.
- The power consumed by AGV starting is proportional to the weight of the AGV load, and the power consumed by the AGV driving is proportional to the weight of the AGV load and the distance traveled by the AGV.
- AGV has the same loading time at each mission point.
- Not consider path conflicts between AGVS.
- Each AGV starts from the same starting point and finally returns to the same target point, each AGV has the same initial power.

2.2. Notations

$K$—Number of AGVs that can be operated

$B$—AGV battery power

$w^i_k, b^i_k$—AGV load weight and battery power when performing the first task $i$

$AGV_k = \{w^i_k, b^i_k | k = 1, 2, \ldots, K\}$—Basic vehicle condition when the $k$-th AGV performs the $j$-th task

$N$—Number of tasks

$r_n = \{x_n, y_n, t^t_n, t^l_n, p_n | 1, 2, \ldots, N\}$—The abscissa, ordinate, generation time, deadline, weight of the $n$-th transportation task

$R_i = \{r_j | r_j \in r_n \}$—Transportation task in the $i$-th task group

$d^t_{AGV_k}$—The total distance traveled by the $k$-th AGV after performing all tasks in task group $i$ $d_{i,j}$—Manhattan distance between task $i$ and task $j$

$i^R_{AGV_k}$—The total time spent by the $k$-th AGV to perform all tasks in task group $i$

$t^{t}_{AGV_k}$—The time spent by the $k$-th AGV from task point $i$ to task point $j$

$q^t_{AGV_k}$—The total power consumed by the $k$-th AGV after executing all tasks in task group $i$

$q^{i,j}_{AGV_k}$—The power consumed from task $i$ to task $j$ when the $k$-th AGV performs the $i$-th task

$g^{i}_{AGV_k}$—The power consumed by the AGV when the $k$-th AGV performs the $i$-th task $start_p, end_p$—Starting point and target point $\alpha, \beta$—constant

Decision variables:

$x_{i,k} = \begin{cases} 1 & \text{Task group } i \text{ is executed by the } k\text{-th AGV} \\ 0 & \text{other} \end{cases}$ (1)

$y_{i,j,k} = \begin{cases} 1 & \text{The } k\text{-th AGV from task point } i \text{ to task point } j \\ 0 & \text{other} \end{cases}$ (2)

Objective function:

$\min Z_1 = \sum_{i=1}^{M} \sum_{k=1}^{K} x_{i,k} q^{R_i}_{AGV_k}$ (3)

$\min Z_2 = \max(x_{i,k} t^{R_i}_{AGV_k})$ (4)

Restrictions:

$q^{R_i}_{AGV_k} = \sum_{i=1}^{m} g^{i}_{AGV_k} + \sum_{i=1, j=1}^{N} y_{i,j,k} q^{i,j}_{AGV_k}$ (5)
3. Improved NSGA-II algorithm

Since the model is a double NP problem, the outer NP determines which tasks the AGV chooses to execute, and the inner NP determines the order of task execution. Therefore, this article improves the NSGA-II algorithm from three aspects.

3.1. Population selection mechanism

The population screening mechanism is a mechanism to eliminate unqualified individuals in the population. It does not need to calculate the constraint violation value of each individual, but only needs to judge according to the number of tasks received by each AGV. The specific steps are as follows:

1. Calculate the number of tasks received by each AGV in the individual.
2. If the number of tasks received by each AGV is less than \( n \), the individual will not be eliminated, otherwise go to step 3.
3. Find all AGVs that have received more tasks in the individual.
4. A certain number of tasks are randomly selected from unqualified AGVs, and these tasks are reassigned to qualified AGVs until the number of tasks in the unqualified AGV is less than \( n \), and the number of tasks in the qualified AGV cannot be greater than \( n \) when the tasks are reassigned.
3.2. **Pheromone-based crossover**

If we use the traditional crossover method (random single-point crossover or random multi-point crossover), the diversity of the population can be improved, but it will reduce the convergence speed of the algorithm and destroy the combination of excellent gene fragments in excellent individuals (the same number) gene fragment. Since the combination of gene fragments in excellent individuals must be select, in order to retain the excellent combination of gene fragments in excellent individuals while improving the quality of the population solution, the concept of pheromone in the ant colony algorithm is introduced, and a pheromone-based crossover method is proposed. This crossover method can well retain the excellent gene fragment combinations in excellent individuals. The specific algorithm flow is as follows:

1. Create a list task for each AGV to record the corresponding task sequence.
2. Group the genes in excellent individuals by AGV number, and store the grouped task sequence in the corresponding list task. If the task sequence exists in the corresponding list, do not add it, otherwise add the task sequence to the corresponding list.
3. Calculate and update the pheromone value of each task sequence.
4. Select the corresponding task sequence by way of roulette.
5. The individuals to be crossed will be rearranged according to the task sequence obtained in step 4 to obtain a new individual.

3.3. **"Double mutation" operator**

Although excellent gene fragment combinations can be retained based on pheromone crossover, it will reduce the population diversity and make the algorithm easy to fall into the local optimal solution. In order to improve the population diversity and avoid the "premature" phenomenon of the algorithm, the "double mutation" algorithm is proposed. The first mutation operator is used to change the position of the excellent gene fragment combination, and the second mutation operator is used to change the number of numbers in the individual. Due to the particularity of the "multi-task" combination model, the diversity of the population mainly depends on the first mutation operator, the second mutation operator assist the first mutation operator so that it can find a better solution. The specific process of the "double mutation" operator is as follows:

1. Randomly select a position from the individual and mark it as \( i \), and the corresponding value at position \( i \) is \( v_i \).
2. Randomly select a position \( j \), \( j \neq i \) and \( v_j \neq v_i \).
3. Exchange the values at positions \( i \) and \( j \), that is, the value at position \( i \) is \( v_j \), and the value at position \( j \) is \( v_i \).
4. Randomly select a position \( k \) from the individual, and make \( v_k = a \) with probability \( p \), where \( a \) is an AGV number whose number of AGV receiving tasks is less than \( n \).

3.4. **Improved NSGA-II algorithm flow**

The improved NSGA-II algorithm flow is as follows:

1. Initialize the population of size \( N \).
2. Use population screening mechanism to eliminate inferior individuals in the initial population.
3. Calculate the objective function value and constraint violation value of each individual.
4. Perform constrained non-dominated sorting based on the individual objective function value and the violation of the constraint value, and calculate the individual’s crowdedness.
5. Set \( Gen \) to 1, start iteration.
6. Select excellent individuals and calculate the pheromone value of excellent gene fragment combinations in excellent individuals.
7. Cross and mutate excellent individuals to generate progeny populations (N).
8. Combine parent and offspring populations (2N).
9. Calculate the objective function value and constraint violation value of the merged population.
10. Perform constrained non-dominated sorting and crowding degree calculation on the population to generate a new population (2N).
11. Use population screening mechanism to eliminate inferior individuals in new populations.
12. If the number of iterations is greater than the set value, output the result, otherwise go to step 6.

4. Experimental simulation

| Task sequence | x coordinate | y coordinate | Task weight | Task deadline |
|---------------|--------------|--------------|-------------|---------------|
| 1             | 15           | 47           | 16          | 75            |
| 2             | 15           | 19           | 1           | 180           |
| 3             | 50           | 35           | 19          | 83            |
| 4             | 22           | 22           | 18          | 48            |
| 5             | 27           | 69           | 10          | 64            |
| 6             | 55           | 60           | 16          | 117           |
| 7             | 41           | 37           | 16          | 59            |
| 8             | 63           | 23           | 2           | 156           |
| 9             | 22           | 27           | 11          | 155           |
| 10            | 25           | 21           | 12          | 153           |
| 11            | 30           | 5            | 8           | 81            |
| 12            | 19           | 21           | 10          | 78            |
| 13            | 22           | 27           | 11          | 155           |
| 14            | 6            | 68           | 30          | 128           |
| 15            | 56           | 37           | 6           | 198           |
| 16            | 15           | 10           | 20          | 62            |
| 17            | 24           | 12           | 5           | 55            |
| 18            | 22           | 22           | 2           | 48            |
| 19            | 56           | 37           | 6           | 198           |
| 20            | 22           | 27           | 11          | 155           |
| 21            | 49           | 73           | 25          | 147           |
| 22            | 55           | 45           | 13          | 136           |
| 23            | 16           | 22           | 41          | 111           |
| 24            | 26           | 27           | 27          | 120           |
| 25            | 56           | 39           | 36          | 162           |
| 26            | 49           | 42           | 13          | 93            |
| 27            | 49           | 11           | 18          | 89            |
| 28            | 60           | 12           | 31          | 64            |
| 29            | 49           | 73           | 25          | 147           |
| 30            | 5            | 30           | 2           | 177           |

Using the R105 data set in the solomon standard data set as the basic experimental data, 30 groups of tasks were randomly selected. Each task includes task location coordinates, task location coordinates,
task generation time, task deadline, and task weight. For specific task data, see Table 1.

Assuming that there are 7 AGVs in the automated logistics system that can be used to perform transportation tasks, each AGV can accept a maximum number of tasks of 5, maximum load $W$ of 100, initial power $B$ of 1000, and driving speed $v$ of 20 (during the AGV driving, the speed remains the same). The coefficient $\alpha$ of the electric energy consumed during the starting process is 0.003, the coefficient $\beta$ of the electric energy consumed during the driving process is 0.001, the AGV loading time $T$ is 5 (the loading time of each task is the same), and the starting point coordinate $start_p$ is (1,100), the end point coordinate $end_p$ is (100,100), the related settings of the algorithm parameters: the population size is 100, the maximum number of iterations is 200, the selection rate is 0.5, the crossover rate is 0.9, the mutation rate is 0.1, the penalty function amplification factor is 2, penalty function offset factor is 1.

The "multi-task" model is simulated and solved in the Matlab environment. The simulation results of the NSGA-II algorithm and the improved NSGA-II algorithm are shown in fig1,2, from fig1,2. It can be seen that the quality of the Pareto optimal solution of the improved NSGA-II algorithm is significantly better than that of the NSGA-II algorithm. In terms of power consumption indicators, the improved algorithm significantly reduces power consumption, and the quality of its solutions is greatly improved.

![Figure 1: NSGA-II algorithm simulation results](image1)

![Figure 2: Improved simulation results of NSGA-II algorithm](image2)

The change process of the Pareto solution of each generation of the NSGA-II algorithm and the change process of the Pareto solution of each generation of the improved NSGA-II algorithm are shown in fig3,4, from fig3,4. The following conclusions can be drawn:

At the beginning of the iteration, the Pareto optimal solution of the original NSGA-II algorithm was few, and the Pareto optimal solution did not change for a long time, indicating that the NSGA-II algorithm was trapped in the local optimal solution during this period. At the beginning of the iteration, the improved NSGA-II algorithm had more types of Pareto optimal solutions, and most of the final Pareto optimal solutions were found in the middle of the iteration, and the number of iterations of inferior individuals accounted for less of the total number of iterations. Compared with the original NSGA-II algorithm, the improved NSGA-II algorithm has more types of Pareto optimal solutions, and most of the final Pareto optimal solutions can be found in the middle of the iteration with a fast search speed. In addition, the optimal solution of Pareto in each generation is almost better than the original NSGA-II algorithm in terms of energy consumption.
In order to eliminate accidental factors and improve the accuracy of the experiment, the above experiment was repeated 20 times with the parameters unchanged to obtain the distribution of the final Pareto optimal solution of the NSGA-II algorithm and the improved NSGA-II algorithm, as shown in fig5. As can be seen from fig5, the original NSGA-II algorithm obtained the final Pareto optimal solution distribution in 20 repeated experiments is relatively discrete, that is, the original NSGA-II algorithm is unstable, while the improved NSGA-II algorithm obtained the final Pareto optimal solution distribution in 20 repeated experiments is relatively concentrated, that is, the improved NSGA-II algorithm is stable. And the quality of the solution is better than the original NSGA-II algorithm.

Figure 5: Pareto optimal solution distribution diagram of the two algorithms repeated 20 times

5. Conclusion
In this paper, the problem of energy consumption in multi-task model and the solving problem of multi-task model are studied. The higher the energy consumption of AGV, the more times of charging during
operation, which will cause system interruption and thus reduce the picking efficiency. In this paper, a multi-task model which minimizes the total energy consumption of AGV and the total time of task out of storage is established by considering the factors of AGV start-up and load. Then, this paper improves the NSGA-II algorithm to solve the model from the following three aspects: population screening mechanism, pheromone-based crossover, and double mutation operate. The experimental simulation results show that the improved NSGA-II algorithm improves the quality of the solutions, and meanwhile improves the stability of the algorithm compared with the original NSGA-II algorithm.

References
[1] Karimi B, Niaki S T A, Haleh H, et al. Bi-Objective optimization of a job shop with two types of failures for the operating machines that use automated guided vehicles. Reliability Engineering & System Safety. 2018; S0951832017308335.
[2] Huang D, Zhang G, editors. Scheduling control of AGV system based on game theory. 6th International Conference on Advanced Infocomm Technology (ICAIT); 2013; Hsinchu, Taiwan: IEEE.
[3] Saidi-Mehrabadi M, Dehnavi-Arani S, Evazabadian F, Mahmoodian V. An Ant Colony Algorithm (ACA) for solving the new integrated model of job shop scheduling and conflict-free routing of AGVs. Computers & Industrial Engineering. 2015; 86:2–13.
[4] Kabir, Q.S.; Suzuki, Y. Comparative analysis of different routing heuristics for the battery management of automated guided vehicles. Int. J. Prod. Res. 2018, 57, 624–641.
[5] Abderrahim M, Bekrar A, Trentesaux D, et al. Manufacturing 4.0 Operations Scheduling with AGV Battery Management Constraints[J]. Energies, 2020, 13(2020):4948.
[6] Kabir, Q.S.; Suzuki, Y. Comparative analysis of different routing heuristics for the battery management of automated guided vehicles. Int. J. Prod. Res. 2018, 57, 624–641.
[7] Mousavi M, Yap HJ, Musa SN, Tahriri F, Md Dawal SZ. Multi-objective AGV scheduling in an FMS using a hybrid of genetic algorithm and particle swarm optimization. PLoS ONE. 2017; 12(3):e0169817. pmid:28263994.
[8] Yaqi ZHANG, Bin YANG, Zhihua HU, et al. Research of AGV charging and job integrated scheduling at automated container terminal. Computer Engineering and Applications. 2017; 53(18):257–262.
[9] Liu Y, Ji S, Su Z, et al. Multi-objective AGV scheduling in an automatic sorting system of an unmanned (intelligent) warehouse by using two adaptive genetic algorithms and a multi-adaptive genetic algorithm[J]. PLOS ONE, 2019, 14.