International Journal of Industrial Engineering Computations 14 (2023) 49–64

Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

Multi-decision points model to solve coupled-task scheduling problem with heterogeneous multi-AGV in manufacturing systems

Xingkai Wang[a,b], **Weimin Wu**[a,b*] **and Zichao Xing**[a,b]

[a] *State Key Laboratory of Industrial Control Technology Institute of Cyber-Systems and Control, Hangzhou, China*
[b] *Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, 310027, China*

CHRONICLE

Article history:
Received August 10 2022
Received in Revised Format
August 31 2022
Accepted October 24 2022
Available online
October, 24 2022

Keywords:
Automated Guided Vehicles (AGVs)
Coupled task scheduling problem
Multi-decision point model
Simulated annealing algorithm

ABSTRACT

Automated guided vehicle (AGV) is widely used in automated manufacturing systems as a material handling tool. Although the task scheduling problem with isomorphic AGV has remained a very active research field through the years, too little work has been devoted to the task scheduling problems with heterogeneous AGVs. A coupled task with heterogeneous AGVs is a complex task that needs the cooperation of more than one type of AGVs. In this paper, a manufacturing system with two types of AGVs and three types of tasks is studied. To solve the coupled task scheduling problem with heterogeneous AGVs in this manufacturing system, we introduce two new methods based on the established mathematical model, namely, the decoupled scheduling strategy and coupled scheduling strategy with multi-decision model. The decoupled scheduling strategy is widely used in coupled task scheduling problems. However, there are some situations that the decoupled scheduling strategy cannot solve the problem well. To overcome the problem, the multi-decision point model solves the coupled task scheduling problem without decomposition. In order to ensure the searching speed and searching accuracy, a novel hybrid heuristic algorithm based on simulated annealing algorithm and tabu search algorithm is developed. The simulation experiment results show the proposed coupled scheduling algorithm has priority in coupled task scheduling problems.

1. Introduction

More and more manufacturing systems adopt automated equipment. As an essential part of automated logistics, the application of automated guided vehicles (AGV) has undoubtedly improved the automation level of the manufacturing system. However, with the increase in the number and type of AGV, designing an effective AGV task scheduling algorithm becomes more challenging than ever. AGV task scheduling refers to assigning tasks to one or multiple AGVs. The AGV task scheduling problem belongs to the robot scheduling problem (RSP), which is always attractive to many scholars. As effective transportation tools, AGVs are introduced to many automated scenarios, such as automated terminals and automated warehouses (Liu & Ioannou, 2002; Singh et al., 2011; Cheng et al., 2005). The AGV scheduling problem is also researched on these scenarios firstly. To solve the problem of dispatching AGV in the changing environment of automated container terminals, Choe et al. propose an online preference learning algorithm (Choe et al. (2016)). In their study, the authors summarize nine parameters that affect AGV dispatching. Each parameter has an effect weight, which will be iteratively updated to ensure the algorithm's robustness. The simulation experiments verify that the proposed algorithm is superior in robustness. Some scholars have proposed simulation-based AGV task scheduling algorithms. Zhicheng et al. (2019) establish a mathematical model based on Petri nets to simulate all operations in the automated container terminal. The travel time and other parameters of AGV will be estimated in the simulation model, which will be adopted to the AGV task scheduling algorithm to promote its accuracy. In order to demonstrate the superiority of battery-powered AGV (B-AGV) to diesel-powered AGV (D-AGV) in automated container terminals, simulation experiments are carried out in terms of charging

* Corresponding author
E-mail: wmwu@iipc.zju.edu.cn (W. Wu)
ISSN 1923-2934 (Online) - ISSN 1923-2926 (Print)

doi: 10.5267/j.ijiec.2022.10.003

strategy and charging station layout by Ma et al. (2021).
In recent years, more and more AGVs are introduced to the manufacturing system to improve the flexibility of the whole system. Different from the AGV scheduling problem in automated container terminals, the cooperation between devices and the collisions of AGVs need to be paid more attention in the manufacturing system due to the limitation in working space of AGVs. Therefore, the multi-objective algorithm has always been one of the focuses of AGV scheduling problems in manufacturing systems. In order to coordinate multiple factors, algorithms such as the fuzzy control algorithm and the multi-stage algorithm are introduced by scholars (Ho et al., 2012; Umashankar & Karthik, 2006; Heger & Voß, 2019). Because of the limitation of space in the manufacturing system, how to deal with conflicts of multiple AGVs is also worthy of research. Miyamoto et al. establish the mathematical model based on the task relationship graph with capacity to achieve conflict-free (Miyamoto and Inoue (2016)). In Murakami's research, a time-space network is used to deal with dispatching and conflict-free routing problems of a capacitated AGV system (Murakami, 2020). However, with the number of AGV increases, classic search algorithms are hard to search for an optimal solution in a short time. So, more and more scholars pay their attention to the heuristic algorithms. Saidi et al. (2015) propose a two-stage ant colony algorithm to solve the conflict-free routing problem and job shop scheduling problem simultaneously. In addition to solving the AGV conflict problem in the AGV task scheduling algorithm, finding the most suitable AGV number for the manufacturing system can also reduce the conflicts of AGVs. Liu and Ioannou (2002) propose a Petri-net based algorithm to calculate the minimum AGV number for manufacturing systems, which will promote the efficiency of AGV and the balance of machine working time. Another feature of AGV scheduling problems in the manufacturing system is the high requirement for timeliness. However, collision avoidance and the uncertain path of multiple AGVs will cause the uncertainty of AGV's travel time, affecting the accuracy of the scheduling result. Witczak et al. (2019) propose an AGV task scheduling algorithm with time fault tolerance. The model will be updated by the difference between the actual travel time and the predicted travel time, which will promote the accuracy of the dispatching results.

With the emergence of different types of AGV tasks, it is increasingly difficult for a single type of AGV to meet all needs. Heterogeneous agents tasks, which need the cooperation of more than one type of agents, are gaining momentum. Yao et al. research the scheduling problem of multiple types of buses (Yao et al. (2020)). A multi-objective optimization algorithm is proposed in their study, which comprehensively considers car purchase costs, charging station installation costs, and dispatching costs. Cross-docking is an effective strategy for transporting perishable goods. However, heterogeneous vehicles and numerous route combinations make manual scheduling difficult. In order to solve this problem, Shahabi-Shahmiri et al. (2021) propose a new multi-objective mixed-integer programming model to minimize the transportation cost. In Zlot and Stentz's (2005) research, the scheduling problem of coupled tasks that need two types of robots' cooperation is simplified by a task tree based decomposition algorithm. A robot scheduling algorithm based on auction is designed to dispatch the decomposed tasks. The coupled tasks mean the dispatching result of multiple tasks has a coupling relationship. Due to this coupling relationship, classic task scheduling algorithms are difficult or even impossible to solve coupled task scheduling problems (CTSP). Therefore, many novel algorithms are introduced to solve CTSP. A plausible task sequence algorithm is proposed based on the concept of maximal fundamental clusters to solve the CTSP of a single machine (Hwang & Lin, 2011). A genetic algorithm with the minimum rank heuristic algorithm is proposed to solve coupled task scheduling problems (Wang et al., 2020).

The case studied in this paper involves two types of AGVs. They cooperate with each other to complete a complex task. According to the taxonomy proposed in paper (Korsah et al. (2013)), this problem belongs to the complex process [CD],the single-task capacity robot [ST], the multi-robot task [MR], and the instantaneous assignment [IA] problem (CD[ST-MR-IA]). The remainder of this paper is organized as follows. In section 2, the research scenario of this paper is introduced in detail and the mathematical model of the problem is established. In section 3, the main process of decoupled scheduling strategy is proposed. The multi-decision model with a novel hybrid heuristic algorithm is proposed to overcome the shortcomings of decoupled scheduling strategy. In section 4, three groups of simulation experiments are carried out to compare the performance of the proposed algorithms. Conclusion and future research for this study are provided in section 5.

2. Problem description and mathematic model

2.1 Problem description

With the increasing complexity of the modern manufacturing system, the material handling system becomes more complex than ever. In most studies, the AGVs are usually used to transport the semi finished products between each two workstations. The transportation of raw materials and finished products is ignored. In this paper, we study the AGV scheduling problem of a complete process in a manufacturing system.

The studied manufacturing system is divided into two areas, the operation area (area in the red block) and the transportation area (area out of the red block), shown by Fig.1. In the operation area, the workstations are connected into a line, and the semi finished products are processed by workers or machines in each workstation. The raw materials are fed from the beginning of the line, and the finished products are shipped out from the end of the line.

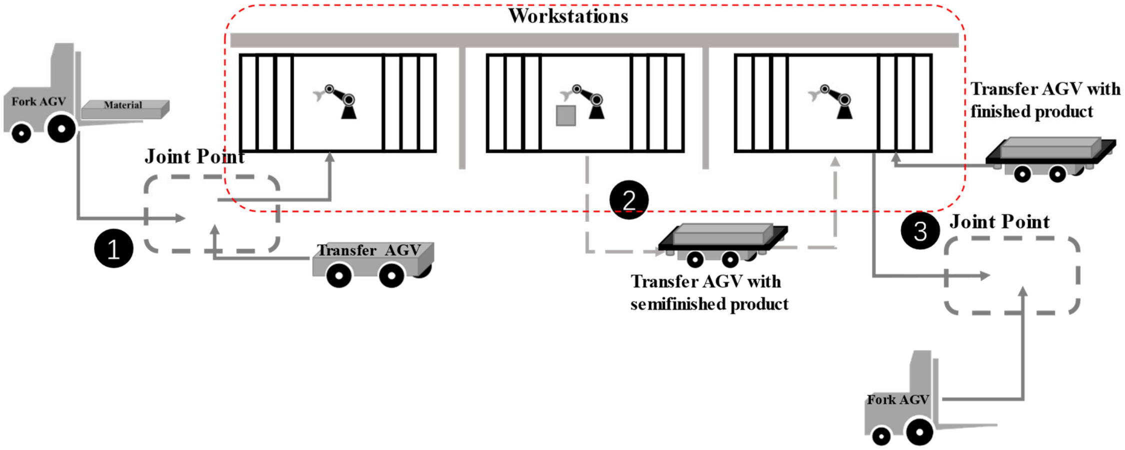
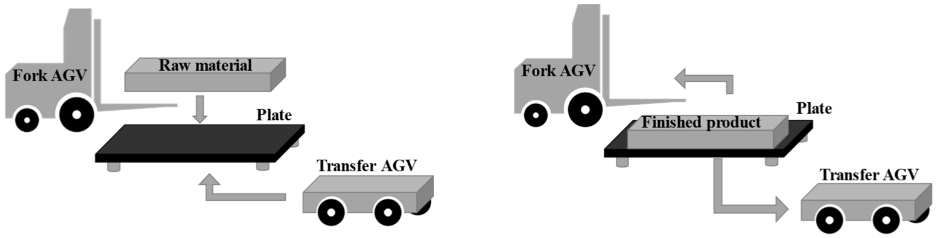


Fig. 1. Transportation of a complete process in the manufacturing system

The transportation of raw materials semi finished products and finished products are performed by two types of AGVs, fork AGV and transfer AGV. Totally three types of transportation tasks are involved in this manufacturing system, the raw material transportation tasks, the semi finished product transportation task and the finished product transportation task. The raw materials transportation tasks are performed by the fork AGV first and then by the transfer AGV, shown as task 1 in Fig.1. The fork AGVs transport the raw materials to the joint point and put the raw materials on the plates. After the raw materials have been laid on the plates, the transfer AGV will come to the joint point and carry the raw materials and plates, shown as Fig.2.(a). Usually, there are multiple joint points in a manufacturing system. Semi finished products are transported between two workstations, which are executed by only transfer AGVs, shown as task 2 in Fig. 1. The last type of task is finished product transportation task, which is opposite to the material transportation task, shown as task 3 in Fig.1. The finished products are transported by the transfer AGVs first, and then by the fork AGV, shown as Fig. 2(b).



(a) Operations at the joint point of task 1 **(b) Operations at the joint point of task 3**

Fig. 2. The operations at the joint point

In this paper, according to the number of AGVs participating in the task, the three types of tasks are divided into two categories. The raw material and finished product transportation tasks are called coupled tasks. The semifinished product transportation tasks are called simple tasks. For the multi-AGV transportation system, the main target is transporting the products as soon as possible, so that the waiting time of each product can be minimized to the minimum value. Therefore, the execution time of each task matters the efficiency of the whole system. Based on this target, the next part establishes the mathematic model of the problem.

2.2 Mathematic model

The notations used in this paper are given in Table 1.

Table 1
Notations

Symbol	Description
$\boldsymbol{\mathcal{A}}$	Set of all AGVs, $\mathcal{A} = \mathcal{AF} \cup \mathcal{AT}$.
$\boldsymbol{\mathcal{AF}}$	Set of fork AGVs, $\mathcal{AF} = \mathcal{AF}_1, \mathcal{AF}_2, \dots, \lvert\mathcal{AF}\rvert$
$\boldsymbol{\mathcal{AT}}$	Set of transfer AGVs, $\mathcal{AT} = \mathcal{AT}_1, \mathcal{AT}_2, \dots, \lvert\mathcal{AT}\rvert$
$\boldsymbol{\mathcal{T}}$	Set of all transportation tasks, $\mathcal{T} = \mathcal{TC} \cup \mathcal{TS}$
$\boldsymbol{\mathcal{TC}}$	Set of coupled tasks, $\mathcal{TC} = \mathcal{TC}^0 \cup \mathcal{TC}^1$
$\boldsymbol{\mathcal{TC}^0}$	Set of raw material transportation tasks.

Table 1

Notations (Continued)

Symbol	Description
$\boldsymbol{\mathcal{TC}^1}$	Set of finished product transportation tasks.
$\boldsymbol{\mathcal{TS}}$	Set of simple tasks, $\mathcal{TS} = \mathcal{TS}_1, \mathcal{TS}_2, \dots, \lvert\mathcal{TS}\rvert$
$\boldsymbol{\mathcal{T}^0_*}$	The last task performed by AGV $*, * \in \mathcal{A}$
$\boldsymbol{N}$	Set of location nodes $N = N_1, N_2, \dots, \lvert N\rvert$
$\boldsymbol{N^j}$	Set of location nodes where the joint point located, $N^j \subset N$.
$\boldsymbol{N_*}$	The starting node when the AGV * is assigned a task, $* \in \mathcal{A}$.
$\boldsymbol{N^s_*}$	The picking node of the task $*, * \in \mathcal{T}$.
$\boldsymbol{N^e_*}$	The target node of the task $*, * \in \mathcal{T}$.
$\boldsymbol{N^m_*}$	The joint point of the coupled task $*, * \in \mathcal{TC}, N^m_* \in N^j$
$\boldsymbol{N^0_*}$	The target point of the last task performed by AGV $*, * \in \mathcal{A}$
$\boldsymbol{F_*}$	The time spent on completing the task $*, * \in \mathcal{T}$.
$\boldsymbol{t^0_{ij}}$	The empty travel time for AGV $\mathcal{A}_i$ performs the task $\mathcal{T}_j$.
$\boldsymbol{t^1_{ij}}$	The loaded travel time for AGV $\mathcal{A}_i$ performs the task $\mathcal{T}_j$.
$\boldsymbol{TT(N_i, N_j)^k}$	The travel time for AGV $\mathcal{A}_k$ from node N_i to N_j.
$\boldsymbol{X(i, j)}$	Decision variable of the AGV $\mathcal{A}_i$ performs the task $\mathcal{T}_j$.

As mentioned above, the optimization target of the studied AGV scheduling problem is to minimize the total makespan of finishing processing with all products. Because the processing time of the products on workstations is an uncontrollable variable, for the AGV scheduling algorithm, the optimization target can be calculated as Eq. (1).

$$minimize: \max_{\mathcal{T}_i \in \mathcal{T}} F_{\mathcal{T}_i} \tag{1}$$

Constraints and calculations are given from Eq. (2) to Eq. (10).

$$F_{\mathcal{TS}_j} = \sum_{\mathcal{AT}_i \in \mathcal{AT}} X(i,j) * \left(t^0_{ij} + t^1_{ij}\right) \tag{2}$$

$$F_{\mathcal{TC}^0_k} = \max\left\{ \sum_{\mathcal{AF}_i \in \mathcal{AF}} X(i,k) * (t^0_{ik} + t^1_{ik}), \sum_{\mathcal{AT}_j \in \mathcal{AT}} X(j,k) * t^0_{jk} \right\} + \sum_{\mathcal{AT}_j \in \mathcal{AT}} X(j,k) * t^1_{jk} \tag{3}$$

$$F_{\mathcal{TC}^1_k} = \max\left\{ \sum_{\mathcal{AF}_i \in \mathcal{AF}} X(i,k) * t^0_{ik}, \sum_{\mathcal{AT}_j \in \mathcal{AT}} X(j,k) * \left(t^0_{jk} + t^1_{jk}\right) \right\} + \sum_{\mathcal{AT}_i \in \mathcal{AF}} X(i,k) * t^1_{ik} \tag{4}$$

$$t^0_{ij} = \begin{cases} TT\left(N_{\mathcal{AT}_i}, N^s_{\mathcal{T}_j}\right)^i, \mathcal{T}_j \in \mathcal{TS} \cup \mathcal{TC}^1, \mathcal{AT}_i \in \mathcal{AT} \\ \mathrm{TT}\left(\mathrm{N}_{\mathcal{AF}_i}, \mathrm{N}^s_{\mathcal{T}_j}\right)^{\mathrm{i}}, \mathcal{T}_j \in \mathcal{TC}^0, \mathcal{AF}_i \in \mathcal{AF} \\ \mathrm{TT}\left(\mathrm{N}_{\mathcal{AT}_i}, \mathrm{N}^m_{\mathcal{T}_j}\right)^{\mathrm{i}}, \mathcal{T}_j \in \mathcal{TC}^0, \mathcal{AT}_i \in \mathcal{AT} \\ \mathrm{TT}\left(\mathrm{N}_{\mathcal{AF}_i}, \mathrm{N}^m_{\mathcal{T}_j}\right)^{\mathrm{i}}, \mathcal{T}_j \in \mathcal{TC}^1, \mathcal{AF}_i \in \mathcal{AF} \end{cases} \tag{5}$$

$$t^1_{ij} = \begin{cases} \mathrm{TT}\left(\mathrm{N}^{\mathrm{s}}_{\mathcal{T}_j}, \mathrm{N}^{\mathrm{e}}_{\mathcal{T}_j}\right)^{\mathrm{i}}, \mathcal{T}_j \in \mathcal{TS} \\ \mathrm{TT}\left(\mathrm{N}^{\mathrm{s}}_{\mathcal{T}_j}, \mathrm{N}^{\mathrm{m}}_{\mathcal{T}_j}\right)^{\mathrm{i}}, \left(\mathcal{T}_j \in \mathcal{TC}^0, \mathcal{AF}_i \in \mathcal{AF}\right) \mathrm{or} \left(\mathcal{T}_j \in \mathcal{TC}^1, \mathcal{AT}_i \in \mathcal{AT}\right) \\ \mathrm{TT}\left(\mathrm{N}^{\mathrm{m}}_{\mathcal{T}_j}, \mathrm{N}^{\mathrm{e}}_{\mathcal{T}_j}\right)^{\mathrm{i}}, \left(\mathcal{T}_j \in \mathcal{TC}^0, \mathcal{AT}_i \in \mathcal{AF}\right) \mathrm{or} \left(\mathcal{T}_j \in \mathcal{TC}^1, \mathcal{AF}_i \in \mathcal{AT}\right) \end{cases} \tag{6}$$

$$\sum_{\mathcal{A}_i \in \mathcal{AT}} X(i,j) = 1, \mathcal{T}_j \in \mathcal{TS} \tag{7}$$

$$\sum_{\mathcal{A}_i \in \mathcal{AT}} \mathrm{X(i,k)} = \sum_{\mathcal{A}_j \in \mathcal{AF}} \mathrm{X(j,k)} = 1, \mathcal{T}_k \in \mathcal{TC} \tag{8}$$

$$\sum_{\mathcal{A}_i \in \mathcal{AT}} X(i,k) = \sum_{\mathcal{A}_j \in \mathcal{AF}} X(j,k) = 1, \mathcal{T}_k \in \mathcal{TC} \tag{9}$$

$$X(i,j) = \begin{cases} 1, AGV\ \mathcal{A}_i\ performs\ task\ \mathcal{T}_j \\ 0, else \end{cases} \tag{10}$$

In above equations, Eq. (2) calculates the time spent on finishing simple tasks. Eq. (3) and Eq. (4) indicate that time spent on finishing a coupled task includes the time spent by the fork AGV and the transfer AGV. The "max" operator calculates the waiting time when the two AGVs interact with each other at the joint point. Eq. (5) calculates the empty travel time when each type of AGV performs different types of tasks. Eq. (6) calculates the loaded travel time when each type of AGV performs different types of tasks. Eq. (7) and Eq. (8) indicate that each simple task can only be assigned to one transfer AGV, but not to fork AGV. Eq. (9) indicates that each coupled task can be assigned to one transfer AGV and one fork AGV. Eq. (10) indicates the value range of the decision variable.

3. Proposed algorithm

3.1 Multi-decision points model

Due to the coupled relationship of multi-type AGVs in the coupled task, it is difficult to solve the coupled task scheduling problem without any pretreatment. In order to solve the coupled task scheduling problem in a method with low complexity. Many scholars put their eyes on the decoupled scheduling algorithms, which means decomposing complex tasks into simple subtasks. This method divided the coupled task scheduling problem into two steps, the coupled task decomposition and the simple task scheduling.

However, when decomposing coupling tasks, most scholars only consider which AGVs to be allocated, which means the other factors may be ignored or be solved in a simple way, such as the joint point selection and the starting time arrangement. For example, in this paper, the decoupled scheduling algorithm to solve the coupled task scheduling problem can be designed as Fig. 3. The joint point selection is solved by choosing the nearest one to the pickup point of the coupled task.

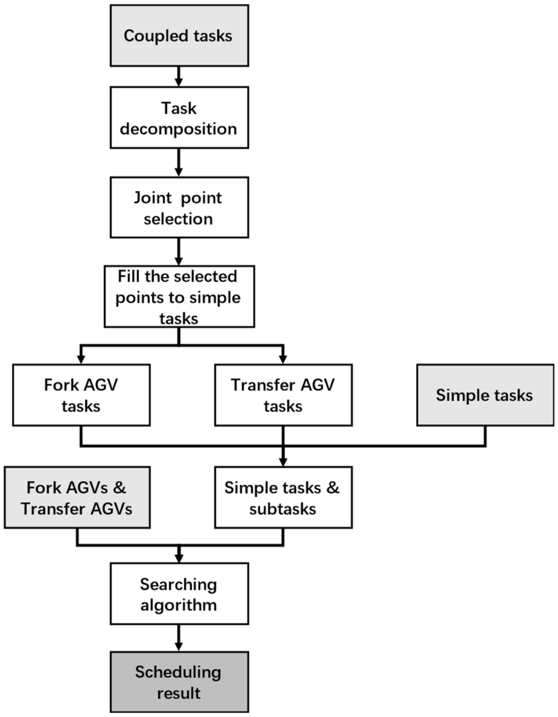


Fig. 3. The flow chart of decoupled scheduling method.

This method may not find the scheduling result with minimum travel distance. For example, Fig. 4 shows the scheduling results of a coupled task. The task transports the finished product from workstation 1 to the storage warehouse. For the decoupled scheduling method, the joint point 2 will be selected for its distance to the pickup point. Therefore, the total transportation distance is 89m, the routes of AGVs are shown by the solid arrows. However, in this case, the best selection is joint point 1 with the total transportation distance of 64m, shown by the dotted arrows.

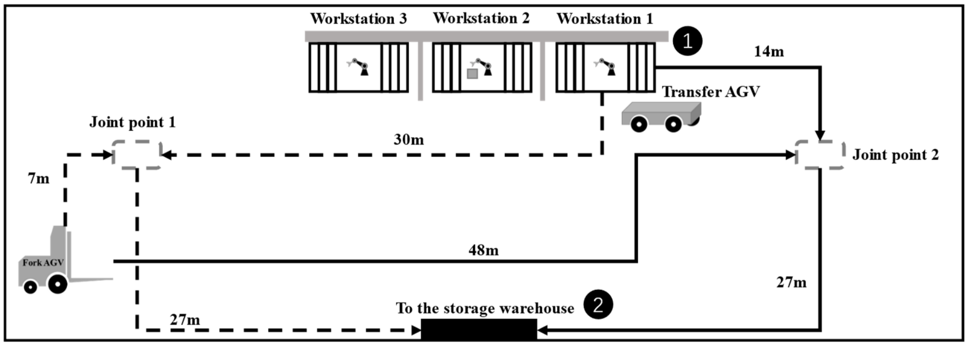


Fig. 4. The shortcoming of decoupled scheduling method.

In order to find the better scheduling result for each coupled task, this paper proposes a multi-decision points method to solve coupled tasks without decomposition. First of all, we give the definition of decision points.

Definition 1: Decision Point. The decision which will influence the cost of the integral coupled AGV task is called a decision point, denoted by DP.

Definition 2: Decision Chain. All decision points belong to a task form a decision chain. The decision chain of a task $\mathcal{T}_j$ is denoted by DC_j, $DC_j = \{DP_j^0, DP_j^1, \dots, DP_j^n\}$.

For example, in this paper, there are three decision points for a coupled task, which are selecting which fork AGV, selecting which transfer AGV and selecting which joint point, shown as Fig. 5. While for a simple task, there is only one decision point.

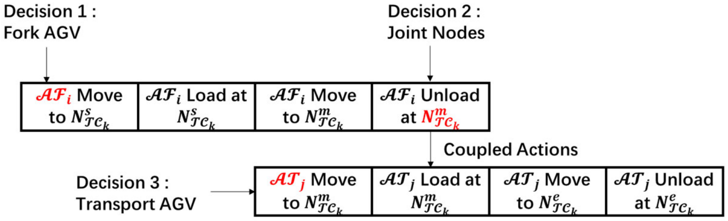
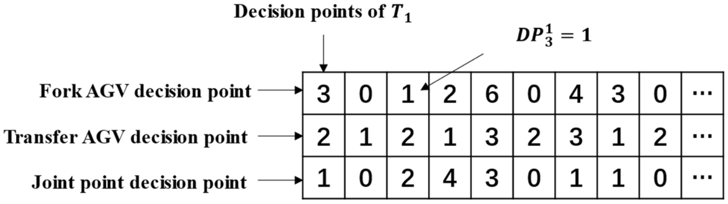


Fig. 5. The decision points of a material transportation task

With the definition on decision point, the solution of a scheduling result can be designed as Fig.6.



Fork AGV decision point	3	0	1	2	6	0	4	3	0	⋯
Transfer AGV decision point	2	1	2	1	3	2	3	1	2	⋯
Joint point decision point	1	0	2	4	3	0	1	1	0	⋯

Fig.6. The decision points of a material transportation task.

In this paper, a solution contains the scheduling result for all simple tasks (semi-finished product transportation task) and all coupled tasks (raw material and finished product transportation tasks). A solution is denoted by Sol_i. Because there are three types of decision points in this problem, the solution is designed with three rows. The first row represents the fork AGV decision point, the second row represents the transfer AGV decision point, and the third row represents the joint point decision point. Each column represents a scheduling result of a task. The numbers in the blocks represent the selected decision point. “0” represents that the decision point is invalid (the fork AGV decision point and the joint point decision point for the simple tasks). For example, the solution shown by Fig.6 represents that the fork AGV and transfer AGV dispatched to the task T_1 are fork AGV 3 and transfer AGV 2. The selected joint point is joint point 1. In this paper, we define the fork AGV decision point as DP_i^1, transfer AGV decision point as DP_i^2 and the joint point decision point as DP_i^3. With this model, the scheduling for coupled tasks can be calculated in one solution without decomposition.

3.2 The hybrid heuristic algorithm

The multi-decision point model builds a $|T| \times 3$ solution. The searching problem for the optimal solution is a NP-hard problem. Because this paper researches the dynamic scheduling problem of coupled tasks, the scheduling solutions have to be found in a short time. Therefore, we proposed a hybrid heuristic algorithm to find the near-optimal solution. The hybrid heuristic algorithm consists with two heuristic algorithms, the simulated annealing algorithm (SA) and tabu search algorithm

(TS) (Zheng et al., 2013).

Simulated annealing algorithm is widely used to solve NP-hard problem for its ability of jumping out of the local optimal solution. However, as a single-agent heuristic algorithm, SA has slower convergence speed than genetic algorithm, particle swarm optimization algorithm and other multi-agent heuristic algorithms. While TS algorithm has good performance in convergence speed, but performs poor in searching result. Therefore, this paper combines these two algorithms to develop a new hybrid heuristic algorithm with good performance in convergence speed and searching result.

The main process of the hybrid heuristic algorithm is shown as Fig.7. The hybrid heuristic algorithm is developed based on classic simulated annealing algorithm, and the tabu search algorithm is integrated to speed up the convergence, shown as gray blocks in Fig.7.

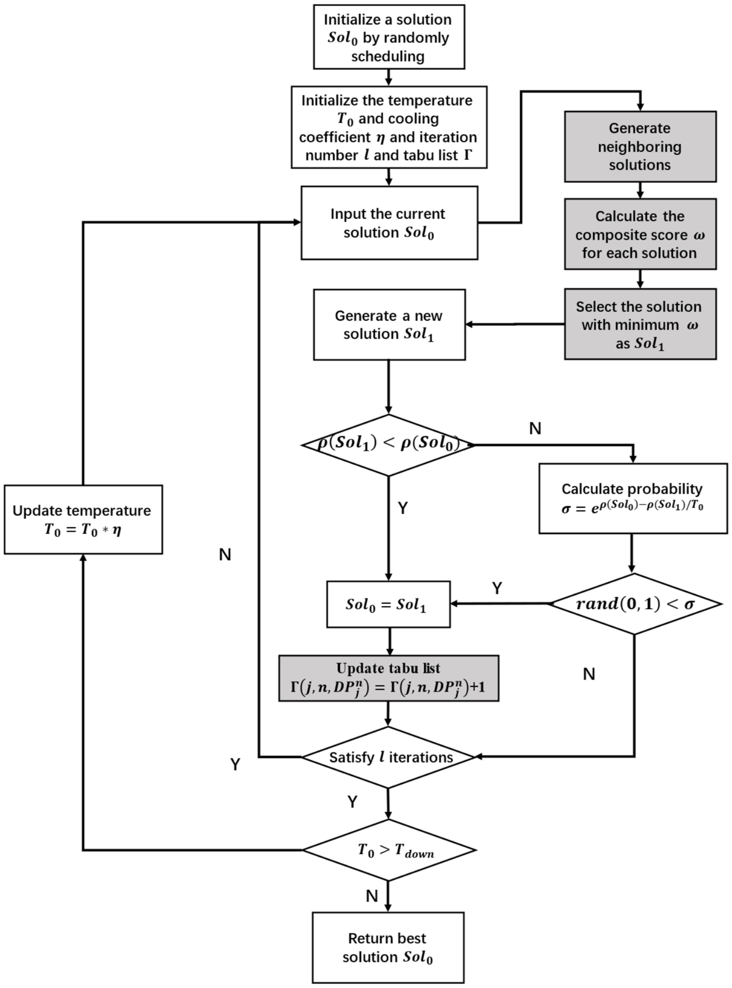


Fig.7. The main process of hybrid heuristic algorithm.

3.3 Solution score

With the solution is designed as Fig.6, the score of the solution is the total completion time of all tasks. The solution score can be calculated as Eq. (11).

$$\rho(Sol_i) = \sum_{\mathcal{T}_j \in \mathcal{T}} \rho_j \tag{11}$$

$$\rho_j = \begin{cases} TT\left(N^0_{\mathcal{AF}_m}, N^s_{\mathcal{T}_j}\right)^m + TT\left(N^s_{\mathcal{T}_j}, N^j_t\right)^m + TT\left(N^0_{\mathcal{AT}_n}, N^j_t\right)^n + TT\left(N^j_t, N^e_{\mathcal{T}_j}\right)^n, \mathcal{T}_j \in \mathcal{TC}^0 \\ TT\left(N^0_{\mathcal{AT}_n}, N^s_{\mathcal{T}_j}\right)^n + TT\left(N^s_{\mathcal{T}_j}, N^j_t\right)^n + TT\left(N^0_{\mathcal{AF}_m}, N^j_t\right)^{mn} + TT\left(N^j_t, N^e_{\mathcal{T}_j}\right)^m, \mathcal{T}_j \in \mathcal{TC}^1 \\ TT\left(N^0_{\mathcal{AT}_n}, N^s_{\mathcal{T}_j}\right)^n + TT\left(N^s_{\mathcal{T}_j}, N^e_{\mathcal{T}_j}\right)^n, \mathcal{T}_j \in \mathcal{TS} \end{cases} \tag{12}$$

$$m = DP_j^1 \tag{13}$$

$$n = DP_j^1 \tag{14}$$

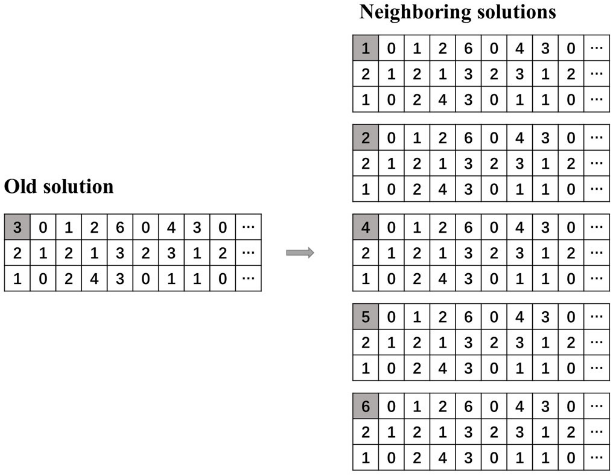
$$t = DP_j^3 \tag{15}$$

where ρ_j represents the score of the task $\mathcal{T}_j$, calculated by Eq.(12). m, n, t represent the selected decision point.

The target of the proposed hybrid heuristic algorithm is finding the solution with minimum value of $\rho(Sol_i)$.

3.3 Neighboring solutions

The neighboring solutions are generated by changing the selected decision point into any other possible decision, shown by Fig.8. Therefore, in this paper, there are $(|\mathcal{AF}|-1)*|\mathcal{TC}|+(|\mathcal{AT}|-1)*(|\mathcal{TC}|+|\mathcal{TS}|)+(|N^j|-1)*(|\mathcal{TC}|)$ new neighboring solutions will be generated in each iteration.



Old solution

3	0	1	2	6	0	4	3	0	···
2	1	2	1	3	2	3	1	2	···
1	0	2	4	3	0	1	1	0	···

Neighboring solutions

1	0	1	2	6	0	4	3	0	···
2	1	2	1	3	2	3	1	2	···
1	0	2	4	3	0	1	1	0	···

2	0	1	2	6	0	4	3	0	···
2	1	2	1	3	2	3	1	2	···
1	0	2	4	3	0	1	1	0	···

4	0	1	2	6	0	4	3	0	···
2	1	2	1	3	2	3	1	2	···
1	0	2	4	3	0	1	1	0	···

5	0	1	2	6	0	4	3	0	···
2	1	2	1	3	2	3	1	2	···
1	0	2	4	3	0	1	1	0	···

6	0	1	2	6	0	4	3	0	···
2	1	2	1	3	2	3	1	2	···
1	0	2	4	3	0	1	1	0	···

Fig. 8. The neighboring solutions generation.

The tabu list is denoted by Γ, which records the number of times each decision is selected. For example, in the first iteration, the retained solution is the solution shown in Fig.6. Then the times of selecting fork AGV 3 for the task 1 is 1. The record is kept by $\Gamma(j, n, DP_j^n)$, which represents the number of times DP_j^n appear in the nth decision point of task $\mathcal{T}_j$. For example, if the in the first iteration, the retained solution is the solution in Fig.6, then $\Gamma(1,1,1) = 0$, $\Gamma(1,1,2) = 0$, $\Gamma(1,1,3) = 1$ and so on. When selecting the new solution from all generated neighboring solutions, the hybrid heuristic algorithm takes composite score to avoid a lot of repeated calculations. The composite score is calculated by Eq. (16).

$$\omega(Sol_i) = \rho(Sol_i) + \alpha * \sum_{DC_j \in Sol_i} \sum_{n=1}^{3} \Gamma(j, n, DP_j^n) \tag{16}$$

where α represents the parameter of duplicate assignment. Therefore, the solution with the minimum composite score will be selected as the new solution. After the solution being selected, records in tabu list Γ will be updated by adding one for each assignment according to the selected solution.

4. Experiments

4.1 Experiment indicators

In this paper, three groups of indicators are used to evaluate the quality of the algorithms. These three groups of indicators are listed in Table 2. For task indicators, transportation distance and execution time are adopted as the indicators. Transportation distance means the total travel distance needed to execute a task, including travel distance of fork AGV and transfer AGV with cargo and without cargo. Execution time means all time spent to complete a task, involving travel time, loading time, and unloading time.

For AGV indicators, the load rate, the utilization rate and the empty time rate are used to evaluate the efficiency of AGVs. The load rate, which LR denotes, is calculated by dividing travel distance with cargo by total travel distance. The higher the load rate, the better the utilization of AGV by the algorithm. UR indicates the utilization rate of the AGVs, which means how busy the AGVs are. A higher utilization rate does not mean AGVs are better utilized. If the other indicators are very close, a lower UR means that the AGVs take less time to complete tasks, which means that the AGVs have higher efficiency. ETR means empty time rate, which is calculated by the ratio of the AGV empty travel time (including waiting) to the total working time of AGV. This indicator evaluates how much time an AGV waits for another AGV when executing the coupled tasks.

Table 2
Experiment Indicators

Group name	Indicators	Description
Task Indicators	ATD	Average transportation distance of all tasks
	AET	Average execution time of all tasks
	CTD	Average transportation distance of all coupled tasks
	CET	Average execution time of all coupled tasks
	STD	Average transportation distance of all simple tasks
	SET	Average execution time of all simple tasks
AGV Indicators	LR	The load rate of all AGVs
	UR	The utilization rate of all AGVs
	ETR	The empty time rate of all AGVs
Processing Indicators	MS	The total makespan of getting 100 finished products

MS is used to indicate the time to get 100 products by the whole system.

4.2 Experiment settings

In this paper, the proposed multi-decision model and hybrid heuristic algorithm are tested in experiments. Each group of experiments is carried out ten times. The simulation runs on a computer with Intel (R) Xeon(R) Platinum 8280L CPU @ 2.60GHz 2.60Hz, and the simulation is programed by Java. Path planning of the AGVs uses the A* algorithm. All experiments start from the idle status of all workstations. All parameters set in the experiments come from the actual production scenarios. The initial temperature and final temperature of SA are set to 99 and 55 respectively. The temperature decline rate is set to 0.95. For the dual consideration of timeliness and effectiveness of the algorithm, the maximum number of iterations is set to 35. AGVs' parameters are listed as Table 3.

Table 3
AGVs' Parameters

Parameter	Value
Acceleration with cargo of fork AGV	0.4 m/s^2
Acceleration without cargo of fork AGV	0.6 m/s^2
Maximum speed with cargo of fork AGV	0.8m/s
Maximum speed without cargo of fork AGV	1m/s
Loading and unloading time of fork AGV	3s
Acceleration with cargo of transfer AGV	0.3 m/s^2
Acceleration without cargo of transfer AGV	0.4 m/s^2
Maximum speed with cargo of transfer AGV	0.8m/s
Maximum speed without cargo of transfer AGV	1m/s
Loading and unloading time of transfer AGV	6s

According to the processing situation of the actual manufacturing system, the number of coupled tasks is far less than that of simple tasks, which means that the number of fork AGVs is far less than that of transfer AGVs. In order to research the impact of the number of fork AGVs on the scheduling result of the algorithms, the manufacturing system is set as Fig. 9. The whole system consists of ten production lines, which share all fork AGVs. While the transfer AGVs are limited in each production line.

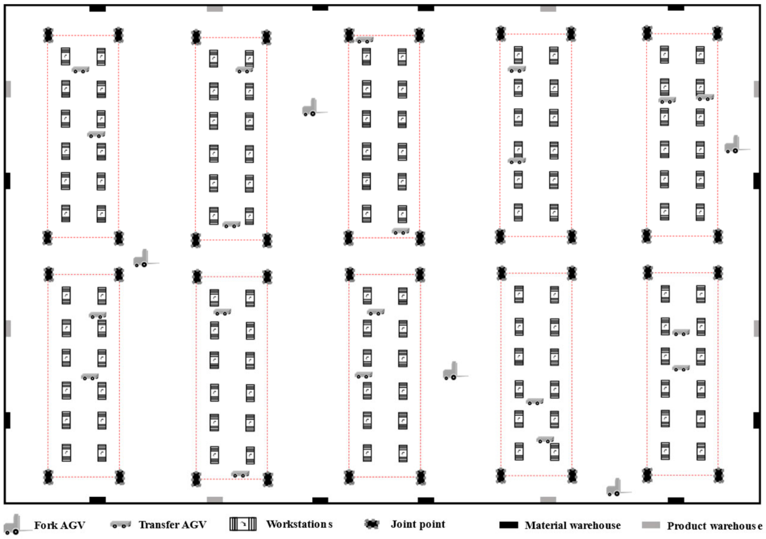


Fig. 9. The layout of experimental manufacturing system

4.3 Experiment on benchmark

In view of the fact that there are few studies on the coupled AGV scheduling problem, we compare the proposed hybrid heuristic algorithm (SATS) with reported algorithms on the benchmark of a similar coupled scheduling problem.

Table 4
Results of experiments on benchmark (The bold is the best value among three algorithms)

Problem	**Searching result**			**Searching time(s)**			**Deviation to the reported value**			**Reported value**
	SA	SATS	TS_SPMA	SA	SATS	TS_SPMA	SA	SATS	TS_SPMA	
EX11	159.0	99.0	**96.0**	23.0	**1.8**	18.6	65.63%	3.13%	0.00%	96
EX21	160.0	106.0	**103.2**	31.0	**2.9**	28.7	60.00%	6.00%	3.22%	100
EX31	220.0	105.0	**103.2**	35.0	**3.6**	33.9	122.22%	6.06%	4.26%	99
EX41	266.0	117.0	**115.4**	47.0	**4.8**	47.6	137.50%	4.46%	3.08%	112
EX51	132.0	94.0	**87.7**	23.1	**1.7**	19.4	51.72%	8.05%	0.77%	87
EX61	191.0	123.0	**120.7**	43.7	**4.7**	45.5	61.86%	4.24%	2.26%	118
EX71	172.0	**118.0**	118.4	43.6	**8.8**	61.4	54.95%	6.31%	6.71%	111
EX81	225.0	**161.0**	**161.0**	42.9	**5.4**	57.5	39.75%	0.00%	0.00%	161
EX91	185.0	123.0	**117.1**	30.9	**3.4**	34.9	59.48%	6.03%	0.96%	116
EX101	278.0	161.0	**150.8**	46.6	**5.9**	65.4	90.41%	10.27%	3.27%	146
EX12	150.0	**82.0**	**82.0**	18.6	**1.8**	18.0	82.93%	0.00%	0.00%	82
EX22	124.0	80.0	**77.8**	25.5	**2.9**	30.1	63.16%	5.26%	2.34%	76
EX32	179.0	86.0	**85.0**	28.7	**3.5**	34.2	110.59%	1.18%	0.00%	85
EX42	238.0	92.0	**90.9**	38.8	**4.4**	47.2	173.56%	5.75%	4.47%	87
EX52	122.0	73.0	**69.0**	17.9	**2.0**	18.3	76.81%	5.80%	0.00%	69
EX62	157.0	102.0	**99.0**	32.4	**4.2**	44.9	60.20%	4.08%	1.02%	98
EX72	136.0	**81.0**	83.9	45.4	**7.3**	62.6	72.15%	2.53%	6.19%	79
EX82	179.0	**151.0**	**151.0**	40.3	**6.2**	58.4	18.54%	0.00%	0.00%	151
EX92	185.0	105.0	**102.6**	29.8	**3.5**	32.3	81.37%	2.94%	0.54%	102
EX102	272.0	138.0	**136.9**	49.3	**6.9**	66.8	101.48%	2.22%	1.40%	135
EX13	153.0	**84.0**	**84.0**	18.8	**2.2**	18.9	82.14%	0.00%	0.00%	84
EX23	130.0	**86.0**	**86.0**	47.0	**3.2**	28.2	51.16%	0.00%	0.00%	86
EX33	182.0	**86.0**	86.3	30.7	**3.7**	37.3	111.63%	0.00%	0.39%	86
EX43	240.0	**93.0**	**93.0**	61.0	**4.6**	49.0	169.66%	4.49%	4.49%	89
EX53	124.0	76.0	**74.6**	18.9	**1.9**	21.1	67.57%	2.70%	0.75%	74
EX63	159.0	105.0	**103.9**	38.7	**4.9**	53.6	54.37%	1.94%	0.86%	103
EX73	142.0	90.0	**88.3**	46.4	**6.9**	72.9	71.08%	8.43%	6.43%	83
EX83	181.0	153.0	**153.0**	46.7	**5.4**	65.2	18.30%	0.00%	0.00%	153
EX93	187.0	**105.0**	106.1	31.1	**3.6**	43.7	78.10%	0.00%	1.06%	105
EX103	278.0	**140.0**	141.2	46.4	**6.0**	74.5	102.92%	2.19%	3.08%	137
EX14	170.0	108.0	**103.2**	17.6	**2.0**	17.3	65.05%	4.85%	0.22%	103
EX24	150.0	117.0	**113.8**	24.1	**3.6**	29.2	38.89%	8.33%	5.35%	108
EX34	238.0	**113.0**	114.2	27.9	**4.3**	35.9	114.41%	1.80%	2.90%	111
EX44	270.0	**129.0**	130.1	42.7	**4.8**	48.5	123.14%	6.61%	7.53%	121
EX54	142.0	98.0	**96.8**	19.2	**2.3**	17.7	47.92%	2.08%	0.81%	96
EX64	199.0	**123.0**	125.9	35.6	**5.1**	44.0	65.83%	2.50%	4.91%	120
EX74	192.0	138.0	**133.6**	45.6	**7.7**	63.2	52.38%	9.52%	6.00%	126
EX84	249.0	**163.0**	**163.0**	44.3	**5.8**	54.9	52.76%	0.00%	0.00%	163
EX94	185.0	**123.0**	123.3	31.3	**3.8**	32.4	54.17%	2.50%	2.78%	120
EX104	298.0	**162.0**	164.3	59.7	**6.5**	65.7	89.81%	3.18%	4.67%	157

Table 4 shows the results on the benchmark proposed by Bilge and Ulusoy (1995). The classic SA and TS_SPMA (the best reported algorithm on the benchmark) are compared in the experiment (Yan et al., 2014). The searching time, searching result and the deviation from the best reported result are used to test the performance of the three algorithms. All experiments were repeated ten times and the results were averaged. Among the three algorithms, the classic SA has the worst performance in terms of searching results and searching time. In terms of searching results, TS_SPMA has the best performance in 27 experiments among 40 experiments, which is the best among the three algorithms. SATS performs best among the three algorithms in 10 experiments. However, in terms of the searching time, SATS has notable advantages. Because the coupled scheduling problem in this paper is solved dynamically, SATS is the best choice for the proposed coupled scheduling strategy due to its good performance on searching speed and searching result.

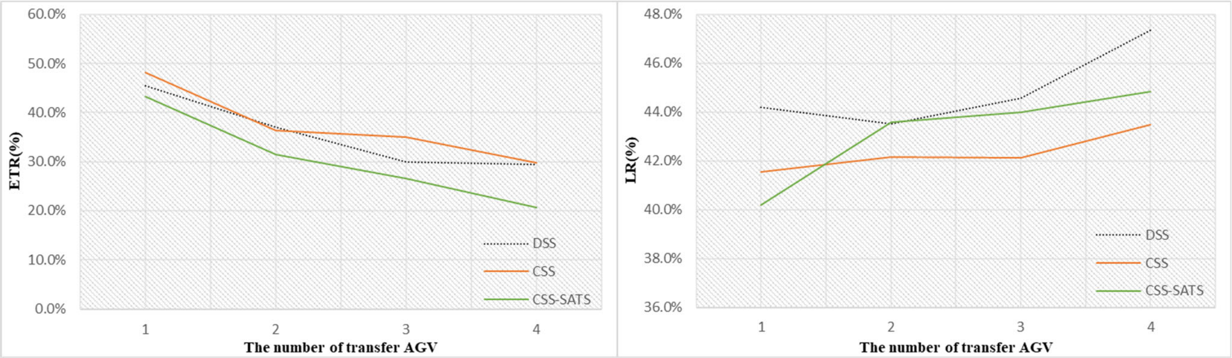
4.4 Experiment A: different transfer AGV number

In this group of experiments, the number of transfer AGV ranges from 10 to 40, which means the number of transfer AGV in each production line ranges from 1 to 4. The processing time of the product on each workstation is set to 15 min. The result is shown as Table 5. Three algorithms are tested in the experiment, the decoupled scheduling strategy (DSS), the coupled scheduling strategy based on multi-decision point (CSS), the coupled scheduling strategy based on multi-decision point and hybrid heuristic algorithm (CSS-SATS). For coupled scheduling strategies without hybrid heuristic algorithms, the classic SA is applied to solve the searching problem.

Table 5
Results of experiment A (The bold is the best value among three algorithms)

Indicators	Transfer AGV number	DSS	CSS	CSS-SATS
LR(%)	1	**44.2%**	41.5%	40.2%
	2	43.5%	42.2%	**43.6%**
	3	**44.6%**	42.1%	44.0%
	4	**47.3%**	43.5%	44.8%
UR(%)	1	69.9%	74.8%	**69.6%**
	2	57.2%	55.9%	**50.4%**
	3	**41.8%**	54.0%	46.9%
	4	**32.7%**	46.1%	45.8%
ETR(%)	1	45.4%	48.1%	**43.3%**
	2	37.1%	36.4%	**31.4%**
	3	29.9%	35.0%	**26.6%**
	4	29.4%	29.8%	**20.6%**
ATD(m)	1	43.6	39.1	**38.1**
	2	42.5	38.5	**37.2**
	3	40.7	36.2	**36.9**
	4	38.7	35.7	**35.5**
CTD(m)	1	167.3	**131.5**	138.5
	2	173.9	145.0	**138.1**
	3	176.5	**146.0**	149.9
	4	174.6	150.2	**144.6**
STD(m)	1	**20.9**	21.2	21.2
	2	**18.7**	19.2	19.0
	3	16.4	16.6	**16.2**
	4	**14.2**	14.8	15.4
AET(s)	1	136.5	130.3	**123.6**
	2	145.2	135.0	**133.2**
	3	222.5	230.1	**199.9**
	4	339.2	336.4	**309.6**
CET(s)	1	370.1	332.3	**304.6**
	2	403.8	346.2	**344.5**
	3	631.5	559.7	**486.2**
	4	975.9	774.7	**719.4**
SET(s)	1	93.7	**90.8**	93.8
	2	98.5	97.0	**94.7**
	3	149.4	171.4	**147.5**
	4	**224.0**	224.8	225.8
MS(s)	1	32786	29292	**28683**
	2	26709	25709	**25447**
	3	35017	30250	**28735**
	4	48629	34485	**34073**

From the result, we can see that the decoupled scheduling strategy (DSS) performs better in terms of AGV load rate, AGV utilization rate and the time spent on simple tasks. This is because the decoupled scheduling strategy completely takes each AGV travel distance as the target, while the coupled scheduling strategy takes the total task transportation distance as the target. That's why the coupled scheduling strategies perform better on coupled task indicators.



(a) **(b)**

Fig. 10. The performance in ETR (a) and LR (b) in experiment A.

Fig. 10(a) shows the performance of the three algorithms in the ETR in experiment A. It can be seen that with the increase of the number of transferred AGVs, ETR is gradually decreasing. This is because the empty time is mainly caused by two situations, the empty travel time of the two kinds of AGVs, and the empty waiting time of the two kinds of AGVs. As the number of transfer AGVs increases, more transfer AGV are available for a task. Therefore, the empty travel distance decreases. This phenomenon is also proved by the increase of load rate, as shown in Fig.11(b). For the empty waiting time of the two kinds of AGVs, when the transfer AGVs are insufficient, the empty time is mainly caused by fork AGVs, which means that fork AGVs need to wait for the transfer of AGVs. Therefore, with the increase of transfer AGVs, the empty waiting time of fork AGVs decreases, resulting in a reduction in the empty time rate.

For the coupled task, it can be seen that the coupled task scheduling algorithm (CSS and CSS-SATS) has obvious advantages in execution time and transportation distance. For CSS and CSS-SATS, we can see that CSS-SATS has obvious advantages in coupled task execution time and empty time rate. The result proves the proposed hybrid heuristic algorithm can search for a better result.

4.5 Experiment B: different fork AGV number

In experiment B, the influence of the number of fork AGV will be tested. The processing time is set to 15 min. The number of fork AGV ranges from 2 to 8. The result of experiment B is shown in Table 6.

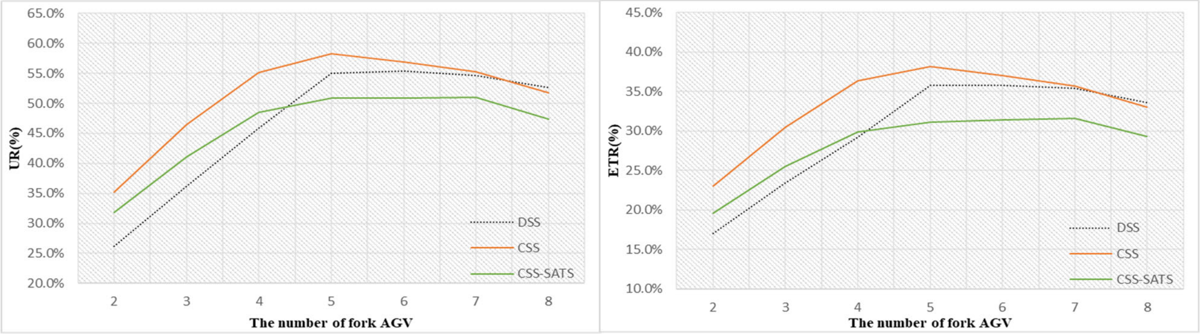
Table 6
Results of experiment B (The bold is the best value among three algorithms)

Indicators	Fork Transfer AGV number	DSS	CSS	CSS-SATS
LR(%)	2	**40.8%**	38.4%	39.7%
	3	**41.1%**	39.2%	38.6%
	4	**42.0%**	39.0%	39.9%
	5	41.0%	40.5%	**42.3%**
	6	**43.0%**	41.8%	42.4%
	7	43.0%	**43.5%**	43.3%
	8	**46.0%**	44.6%	44.8%
UR(%)	2	**26.2%**	35.2%	31.9%
	3	**36.2%**	46.5%	41.1%
	4	**45.8%**	55.1%	48.5%
	5	55.0%	58.3%	**50.9%**
	6	55.4%	56.9%	**50.8%**
	7	54.7%	55.3%	**51.1%**
	8	52.6%	51.7%	**47.4%**
ETR(%)	2	**17.1%**	23.1%	19.6%
	3	**23.4%**	30.5%	25.5%
	4	**29.2%**	36.3%	29.9%
	5	35.8%	38.2%	**31.1%**
	6	35.8%	37.0%	**31.4%**
	7	35.4%	35.7%	**31.6%**
	8	33.6%	33.0%	**29.3%**
ATD(m)	2	45.5	39.8	**38.9**
	3	44.0	40.5	**40.2**
	4	43.8	39.7	**39.4**
	5	43.9	39.9	**37.9**
	6	42.5	38.5	**37.2**
	7	41.7	37.6	**37.1**
	8	40.1	36.6	**36.4**
CTD(m)	2	194.3	160.8	**158.3**
	3	185.6	165.5	**163.9**
	4	186.7	157.0	**156.4**
	5	183.9	156.1	**144.0**
	6	173.9	145.0	**138.1**
	7	165.1	139.5	**135.4**
	8	155.5	133.9	**133.4**
STD(m)	2	18.6	19.3	**18.5**
	3	**18.3**	18.9	18.8
	4	**17.7**	18.7	18.3
	5	**18.6**	18.9	18.7
	6	**18.7**	19.2	19.0
	7	19.3	19.3	**19.0**
	8	18.9	**18.8**	19.0
AET(s)	2	189.1	121.6	**121.4**
	3	163.4	127.8	**127.2**
	4	155.5	131.1	**129.6**
	5	146.6	136.4	**131.4**
	6	139.3	136.9	**135.3**
	7	146.7	141.5	**136.6**
	8	142.8	137.2	**133.5**

Table 6
Results of experiment B (The bold is the best value among three algorithms) (Continued)

Indicators	Fork Transfer AGV number	DSS	CSS	CSS-SATS
CET(s)	2	768.4	327.1	**319.1**
	3	578.6	344.7	**340.1**
	4	522.2	348.6	**345.9**
	5	435.1	360.3	**341.4**
	6	387.9	353.0	**344.5**
	7	394.9	359.3	**343.5**
	8	380.1	347.6	**341.0**
SET(s)	2	**84.2**	87.9	86.6
	3	**88.2**	90.4	90.2
	4	**88.6**	92.8	90.1
	5	94.2	95.8	**93.4**
	6	**94.2**	97.8	97.4
	7	101.6	**98.9**	101.9
	8	99.4	**95.8**	99.2
MS(s)	2	52687	37628	**37227**
	3	38130	30542	**30414**
	4	29954	26534	**26413**
	5	27197	25668	**25578**
	6	26384	25861	**25592**
	7	26901	26190	**25495**
	8	26450	25958	**25235**

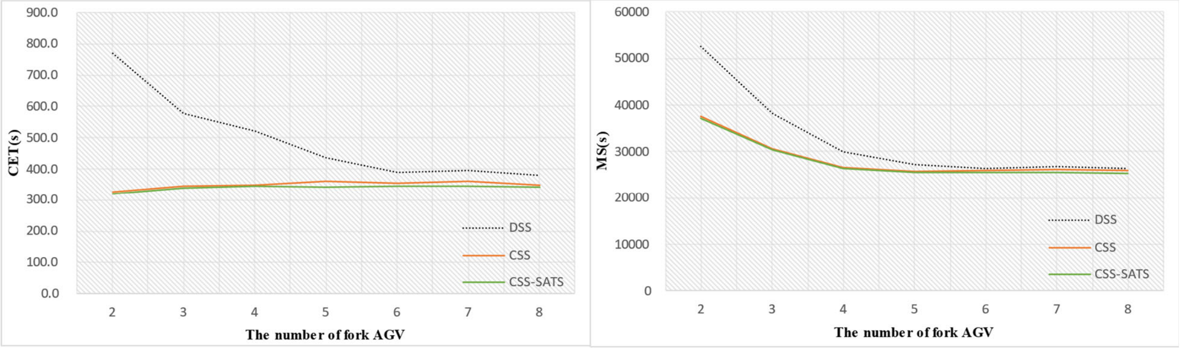
There is little difference between the three algorithms in terms of simple task execution. For utilization rate and empty time rate of AGV, it can be seen that when there are a few fork AGVs, the decoupled scheduling algorithm performs better. With the increase of the number of fork AGVs, the performance of the coupled task scheduling strategy is getting better and better, as shown in Fig.11(a) and Fig.11(b).



(a) (b)

Fig. 11. The performance in UR (a) and ETR (b) in experiment B

It is worth noting that in experiment A, the empty time rate decreased with the increase of transfer AGVs. This is because that when the transfer AGVs are insufficient, the empty time is mainly caused by the waiting time of fork AGVs. This phenomenon is proved by experiment B. As shown in Fig.11(b), with the increase of fork AGVs, the empty time rate increased at first, which is just opposite to experiment A. This result indicates that the empty time is mainly caused by the empty waiting of fork AGV. Therefore, with the increase of fork AGVs, the empty time rate is increasing.



(a) (b)

Fig. 12. The performance in CET (a) and MS (b) in experiment B

However, low utilization rate and low empty time rate do not mean the decoupled scheduling strategy performs better than other two algorithms. In the performance of coupled tasks, we can see that the coupled scheduling strategies have absolute advantages. As shown in Fig.12.(a), in terms of execution time of coupling tasks, the increase of fork AGVs has little impact on coupled scheduling strategy. This phenomenon indicates that the coupled task scheduling strategy can make better use of fork AGVs, and even if the fork AGVs are insufficient, the transportation needs can be met. For the decoupling scheduling strategy, the increase of fork AGVs has a great impact on the task execution time. The same phenomenon also occurs in the product processing time, as shown in Fig.12 (b).

4.6 Experiment C: different processing time

In an intelligent manufacturing system, the processing time on workstations differs when processing different types of products. The experiment C tests the proposed algorithms in cases with different process time. The number of fork AGVs and the number of transfer AGVs are fixed. And the processing time ranges from 3min to 21min. The experimental result of experimental C is shown in Table 7.

Table 7

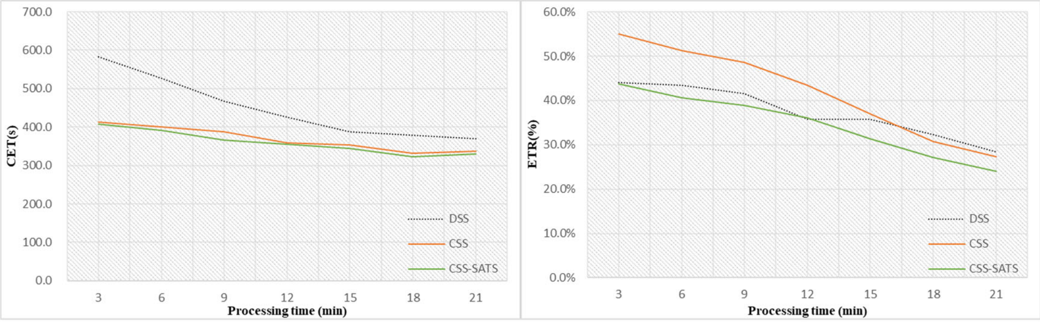
Results of experiment C (The bold is the best value among three algorithms)

Indicators	Processing time (min)	DSS	CSS	CSS-SATS
LR(%)	3	**41.8%**	38.5%	40.0%
	6	**42.0%**	40.1%	41.4%
	9	**42.7%**	40.0%	41.9%
	12	**42.4%**	40.3%	41.7%
	15	**43.0%**	41.8%	42.4%
	18	43.4%	42.6%	**44.3%**
	21	**44.0%**	43.4%	43.5%
UR(%)	3	**69.2%**	83.0%	68.4%
	6	67.8%	78.3%	**66.2%**
	9	65.1%	74.0%	**62.8%**
	12	63.6%	66.3%	**58.2%**
	15	55.4%	56.9%	**50.8%**
	18	50.3%	47.7%	**44.2%**
	21	44.0%	42.3%	**39.0%**
ETR(%)	3	44.1%	55.0%	**43.8%**
	6	43.5%	51.2%	**40.6%**
	9	41.5%	48.5%	**39.0%**
	12	35.8%	43.4%	**36.0%**
	15	35.8%	37.0%	**31.4%**
	18	32.4%	30.7%	**27.1%**
	21	28.4%	27.3%	**24.0%**
ATD(m)	3	43.7	**39.5**	39.9
	6	42.5	**38.9**	39.1
	9	42.2	39.3	**38.1**
	12	43.8	39.2	**38.2**
	15	42.5	38.5	**37.2**
	18	42.6	37.3	**36.5**
	21	41.5	37.5	**37.2**
CTD(m)	3	185.6	**151.0**	152.8
	6	183.2	153.3	**151.1**
	9	180.6	151.4	**148.6**
	12	182.0	147.4	**143.5**
	15	173.9	145.0	**138.1**
	18	173.5	137.7	**134.8**
	21	164.8	139.5	**137.0**
STD(m)	3	**18.9**	19.8	19.6
	6	**17.5**	18.5	18.6
	9	**17.3**	19.0	18.1
	12	**19.0**	19.5	19.2
	15	**18.7**	19.2	19.0
	18	18.9	19.2	**18.5**
	21	19.1	19.1	**19.0**
AET(s)	3	174.5	160.7	**159.3**
	6	163.8	152.5	**148.4**
	9	153.5	145.4	**141.0**
	12	148.8	141.2	**139.1**
	15	139.3	136.9	**135.3**
	18	137.6	130.0	**127.2**
	21	135.3	130.0	**129.4**

Table 7
Results of experiment C (The bold is the best value among three algorithms) (Continued)

Indicators	Processing time (min)	DSS	CSS	CSS-SATS
CET(s)	3	582.6	412.5	**407.3**
	6	527.6	400.8	**392.0**
	9	466.4	387.2	**367.0**
	12	425.5	358.2	**355.7**
	15	387.9	353.0	**344.5**
	18	379.3	332.5	**323.3**
	21	369.3	336.5	**329.5**
SET(s)	3	**103.3**	115.4	115.4
	6	**99.2**	104.9	107.2
	9	**97.1**	100.1	101.8
	12	**99.2**	101.7	99.9
	15	**94.2**	97.8	97.4
	18	**93.9**	91.7	92.9
	21	**93.0**	93.1	92.6
MS(s)	3	17422	17825	**15741**
	6	19411	18847	**17011**
	9	22004	20845	**19160**
	12	23493	22708	**22022**
	15	26384	25861	**25592**
	18	29595	28728	**28717**
	21	33230	32528	**33515**

In the experiment C, we can see that the decoupled scheduling strategy performs better in the results of simple tasks (including the execution time and transportation distance of simple tasks). While, on the performance of coupled tasks (including the execution time and transportation distance of coupled tasks), the coupled scheduling strategies have better results. In terms of the total product processing time, the coupled scheduling strategies also get a better result than the decoupled scheduling strategy.



(a) (b)

Fig. 13. The performance in CET (a) and ETR (b) in experiment C

As for the execution time of coupled tasks, it can be seen that with the increase of processing time, the coupled scheduling strategies change little. However, for the decoupled scheduling strategy, the execution time of the coupled task decreases sharply with the increase of the processing time, as shown in Fig.13.(a). This is because with the increase of processing time, the number of simple tasks is reduced. Therefore, after decomposition, the proportion of fork AGV subtasks to transfer AGVs subtasks has increased, which means that the two subtasks (fork AGV subtasks and transfer AGV subtasks) decomposed by a coupled task have a greater probability of being executed at the same time. Therefore, the time for two types of AGVs to wait for each other is reduced, which can be proved by the empty time rate of experiment C in Fig.13.(b).

5. Conclusion

In the manufacturing system, an efficient scheduling algorithm can improve the system's overall efficiency and reduce AGV costs. In this paper, a coupled task scheduling problem with heterogeneous multi-type AGVs is researched. First of all, a decoupled scheduling method is proposed based on the established mathematical model. However, the decoupled method performs poorly in some situations. In order to obtain a better solution, a multi-decision points model is proposed. A novel hybrid heuristic algorithm based on simulated annealing algorithm and tabu search algorithm is proposed to solve the NP-hard searching problem in the multi-decision point model. Finally, the results of the experiments show the effectiveness of the proposed algorithms. When the number of AGV is not ample, the multi-decision point model can make better use of AGVs than decoupled scheduling strategy. The algorithms proposed in this paper are already applied in an air-conditioning workshop (Fig.14). The future work will focus on how to reinforce the robustness of the algorithm. The accuracy of the scheduling result will become worse due to the uncertainty of AGV travel time in the experiments. The focus of the following research will be to eliminate the impact of the difference between estimated travel time and the actual travel time.

Fig.14. Transfer AGVs (yellow) and fork AGVs (blue)

References

Bilge, Umit & Ulusoy, Gunduz. (1995). A Time Window Approach to Simultaneous Scheduling of Machines and Material Handling System in an FMS. *Operations Research, 43*, 1058-1070.

Cheng, Y. L., Sen, H. C., Natarajan, K., Teo, C. P., & Tan, K. C. (2005). Dispatching automated guided vehicles in a container terminal*. In Supply chain optimization* (pp. 355-389). Springer, Boston, MA.

Choe, R., Kim, J., & Ryu, K. R. (2016). Online preference learning for adaptive dispatching of AGVs in an automated container terminal. *Applied Soft Computing, 38*, 647-660.

Heger, J., & Voß, T. (2019). Dynamic priority based dispatching of AGVs in flexible job shops. *Procedia CIRP, 79*, 445-449.

Ho, Y. C., Liu, H. C., & Yih, Y. (2012). A multiple-attribute method for concurrently solving the pickup-dispatching problem and the load-selection problem of multiple-load AGVs. *Journal of manufacturing systems, 31*(3), 288-300.

Hwang, F. J., & Lin, B. M. (2011). Coupled-task scheduling on a single machine subject to a fixed-job-sequence. *Computers & Industrial Engineering, 60*(4), 690-698.

Korsah, G. A., Stentz, A., & Dias, M. B. (2013). A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research, 32*(12), 1495-1512.

Liu, C. I., & Ioannou, P. A. (2002). A petri net based approach for AGV dispatch scheduling and fleet size determination. *IFAC Proceedings Volumes, 35*(1), 19-24.

Liu, C. I., & Ioannou, P. A. (2002, September). A comparison of different AGV dispatching rules in an automated container terminal. In Proceedings. *The IEEE 5th International Conference on Intelligent Transportation Systems* (pp. 880-885). IEEE.

Ma, N., Zhou, C., & Stephen, A. (2021). Simulation model and performance evaluation of battery-powered AGV systems in automated container terminals. *Simulation Modelling Practice and Theory, 106*, 102146.

Miyamoto, T., & Inoue, K. (2016). Local and random searches for dispatch and conflict-free routing problem of capacitated AGV systems. *Computers & Industrial Engineering, 91*, 1-9.

Murakami, K. (2020). Time-space network model and MILP formulation of the conflict-free routing problem of a capacitated AGV system. *Computers & Industrial Engineering, 141*, 106270.

Saidi-Mehrabad, M., Dehnavi-Arani, S., Evazabadian, F., & Mahmoodian, V. (2015). An Ant Colony Algorithm (ACA) for solving the new integrated model of job shop scheduling and conflict-free routing of AGVs. *Computers & Industrial Engineering, 86,* 2-13.

Shahabi-Shahmiri, R., Asian, S., Tavakkoli-Moghaddam, R., Mousavi, S. M., & Rajabzadeh, M. (2021). A routing and scheduling problem for cross-docking networks with perishable products, heterogeneous vehicles and split delivery. *Computers & Industrial Engineering, 157*, 107299.

Singh, N., Sarngadharan, P. V., & Pal, P. K. (2011). AGV scheduling for automated material distribution: a case study. *Journal of Intelligent Manufacturing, 22*(2), 219-228.

Umashankar, N., & Karthik, V. N. (2006, June). Multi-criteria intelligent dispatching control of automated guided vehicles in FMS. *In 2006 IEEE Conference on Cybernetics and Intelligent Systems (pp. 1-6). IEEE.*

Wang, H., Chen, W., & Wang, J. (2020). Coupled task scheduling for heterogeneous multi-robot system of two robot types performing complex-schedule order fulfillment tasks. *Robotics and Autonomous Systems, 131*, 103560.

Witczak, M., Majdzik, P., Stetter, R., & Lipiec, B. (2019). Multiple AGV fault-tolerant within an agile manufacturing warehouse. *IFAC-PapersOnLine, 52*(13), 1914-1919.

Yao, E., Liu, T., Lu, T., & Yang, Y. (2020). Optimization of electric vehicle scheduling with multiple vehicle types in public transport. *Sustainable Cities and Society, 52*, 101862.

Zheng, Y, Xiao, Y & Seo, Y. (2013). A tabu search algorithm for simultaneous machine/AGV scheduling problem. *International Journal of Production Research, 52*, 5748-5763.

Zhicheng, B., Yaozhou, Z., Xuemin, Z., Yansong, X., Jiaqi, C., & Weijian, M. (2019, May). Simulation-based AGV dispatching in automated container terminal. In 2019 *International Conference on Advances in Construction Machinery and Vehicle Engineering (ICACMVE) (pp. 414-420).* IEEE.

Zlot, R., & Stentz, A. (2005, April). Complex task allocation for multiple robots*. In Proceedings of the 2005 IEEE international conference on robotics and automation (pp. 1515-1522).* IEEE.