Journal of Manufacturing Systems 57 (2020) 19–30

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

A study for further exploring the advantages of using multi-load automated guided vehicles

Rundong Yan*, Lisa Jackson, Sarah Dunnett

Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, Leicestershire, LE11 3TU, UK

ARTICLE INFO

Keywords:
Logistics
Scheduling
Automated guided vehicle
Petri Net
Simulation

ABSTRACT

Multi-load Automated Guided Vehicle (AGV) has been proposed for a while, but the advantage of its application is still not fully understood today. In order to fill this knowledge gap, a novel integrated model of a multi-load AGV system is developed in this paper by using an advanced form of Petri Nets, namely Coloured Petri Nets (CPNs), to simulate the operation of the AGV system in various scenarios. The study reported in this paper is focused to answer a few key questions, i.e. whether system performance can be continuously improved by increasing the load capacity of the multi-load AGVs; if not, whether there is an optimal load capacity of the multi-load AGV for a particular system; and whether the multi-load AGVs can still work well in a system with flexible loading and unloading points. The research results have shown that the efficiency of the AGV system can be improved by increasing the load capacity at the beginning, but the effectiveness of such an approach will decrease when the load capacity increases above a certain value. In other words, an AGV system may not perform better after using a larger capacity of multi-load AGV and there must be an optimal load capacity of the multi-load AGV for a specific AGV system. In addition, it is found that a system with flexible loading and unloading points can perform better after using a multi-load AGV.

1. Introduction

Automated Guided Vehicles (AGVs) are increasingly used in a variety of areas of modern industry in order to improve efficiency and productivity. Such vehicles run along a predefined route to deliver prescribed tasks without the involvement of an on-board operator. Much effort has been made previously in this area in order to achieve an efficient AGV system and reduce operating costs. However, most of these efforts consider the design and optimisation of the layout, control, and traffic management of single-load AGV systems [1–4]. For example, several studies were conducted to address the conflict and deadlock issues that often occur and cause a decrease in system efficiency [5–7]. However, with the continual scaling up of the AGV systems, more and more AGVs operate simultaneously in the same system, which makes system control and management more difficult than ever before. The research was also conducted for improving the efficiency of complex AGV systems. For example, a genetic algorithm was employed to minimise the unloading time of an AGV system that has 200 containers and up to 10 AGVs in a container terminal [8]; A method was developed to identify the minimum number of AGVs required by an AGV system to guarantee that every job can be immediately taken up by an AGV at its known release time [9]. Besides, the idea of decentralised resource management was discussed in order to further improve the efficiency of AGV systems [10]. However, despite these efforts, the improvement of the efficiency of a large scale AGV system is still limited due to the use of single-load AGVs and their limited load capacities.

In order to address this issue, a new concept AGV, namely a multi-load AGV, was present [11]. As opposed to single-load AGVs, a multi-load AGV is able to pick up multiple items at each station and is also able to pick up items from multiple stations. In order to understand how to control a multi-load AGV to deliver different types of tasks, a single multi-load AGV along a single-loop guide-path and an AGV system containing three four-load AGVs were simulated in [12] and [13], respectively. Both studies suggested that the control strategies, task-determination rules, and delivery-dispatching rules have a significant impact on the performance of the AGV systems. Besides these, the performances of single- and dual-load AGV's in automated seaport container terminals were compared in [14]; the impact of different dispatch rules and the speed of multi-load AGVs on the performance of the AGV system was also investigated in [15]. In addition, a multiple-attribute method was proposed to solve the pickup-dispatching and the load-selection problems in multi-load AGVs [16]. Despite these studies, the optimal use of multi-load AGVs is still not fully understood. For example, it is not clear whether system performance can be

* Corresponding author.
E-mail addresses: r.yan@lboro.ac.uk (R. Yan), l.m.jackson@lboro.ac.uk (L. Jackson), s.j.dunnett@lboro.ac.uk (S. Dunnett).

https://doi.org/10.1016/j.jmsy.2020.08.005
Received 8 December 2019; Received in revised form 24 June 2020; Accepted 4 August 2020
Available online 10 August 2020
0278-6125/ © 2020 The Authors. Published by Elsevier Ltd on behalf of The Society of Manufacturing Engineers. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

continuously improved by increasing the load capacity of the AGVs and if not, whether there is an optimal load capacity of the multi-load AGV for a particular system. In addition, most AGV systems that were considered in previous studies had fixed loading and unloading points. Whether AGVs can still work efficiently in a system with flexible loading and unloading points is also an interesting question to answer. All these questions will be answered in this paper through investigating the impact of load capacity of a multi-load AGV on system performance and its efficiency when operating in a system with flexible loading and unloading points.

In order to get answers to these questions, Petri Nets (PNs) based Discrete Event Simulation (DES) [17] is adopted in this paper to simulate the multi-load AGV systems of interest. The reason for using PNs is that PNs have been proved more flexible and more computationally efficient than conventional DES particularly when modelling complex systems [18–20]. Of course, these advantages of PNs are equally applicable to modelling AGV systems [21]. In fact, PNs have been successfully used in the past years to simulate various issues in AGV systems, such as deadlock problems [22], reliability issues [23,24], and so on. However, despite that the traditional PNs have demonstrated advantages in modelling the AGV systems in which every AGV is requested to deliver only repeated tasks, they meet difficulties when modelling AGV systems that request individual AGVs to undertake different tasks as needed over time. Coloured Petri Nets (CPNs) [25], an extension form of PNs, provide a potential tool for overcoming these issues. Considering that CPNs offer extended flexibility than traditional PNs for the dynamic modelling of more complex AGV systems [5,26], it is employed in this paper to simulate the multi-load AGV systems. The scientific contributions of this research are:

- new and unique CPN models are developed to simulate the operation of AGV systems in different scenarios;
- the impact of the load capacity of multi-load AGV on the efficiency and performance of the whole AGV system is investigated with the aid of the developed CPN models;
- besides conflict, flexible loading and unloading stations are considered, which makes the research results more realistic in describing the operation of AGV systems.

The remaining part of this paper is organised as follows:

The CPN modelling method is briefly introduced in Section 2; the layout of the AGV system considered is described in Section 3; three types of CPN models are developed in Section 4 which take into account the number and load capacity of AGVs, mission allocation, routing problem, and conflict avoidance; the simulation calculations and discussion are conducted in Section 5; the paper ends with the key research conclusions in Section 6.

2. Petri Nets, coloured Petri Net and its improvement

PN provides an intuitive graphical representation of a system and their flexibility enables modifications in design, operation & maintenance of a system to be easily incorporated [27–29]. As shown in Fig. 1, a PN is a direct bipartite graph, which consists of four types of symbols, i.e. circles, rectangles, arrows, and tokens. The circles represent places and the rectangles represent transitions. If the time for

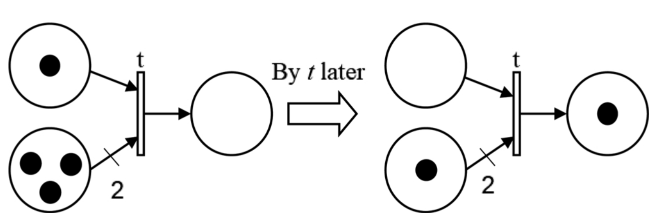


Fig. 1. Petri Net model with transitions.

completing the transition is zero, the rectangle will be filled in black, otherwise, it will not be filled; the arrows represent arcs that connect the places and transitions. The arcs with a slash on and a number ‘n’ next to the slash represent a combination of n single arcs and the arcs are said to have a weight ‘n’. The weight will default to one if there is no slash. Small filled-in circles represent tokens, which carry the information in the PNs. The tokens move via transitions if the enabling condition is satisfied. This defines the dynamic properties of PNs. For example, in Fig. 1, there are one and three tokens in the two input places to the transition. The input places have arcs with weights 1 and 2, respectively. The transition will be enabled once the number of tokens contained in every input place is equal to, or greater than, the corresponding arc weights. Hence, the transition in Fig. 1 is enabled and consequently, one token will be transferred to the output place after the switching time ‘t’ that is associated with the transition. The number of tokens that will be produced in the output place is dependent on the ‘weight’ of the corresponding arc. If the arc weight is ‘n’, then ‘n’ tokens will appear in the output place after the transition is completed. The number of tokens in the input places corresponding to the arc weights are removed. To ease understanding of the aforementioned transition process, an example is given in Fig. 1. The transition can be activated only when both enabling conditions are satisfied, i.e. there must be at least one token in the top input place and at least two tokens in the bottom input place as the corresponding weights of the arcs connected to them are one and two, respectively. Since the transition is associated with a time delay t, the transition will fire after time t. Consequently, one token is produced in the output place as the weight of the arc connecting the transition and the output place is defaulted to be one.

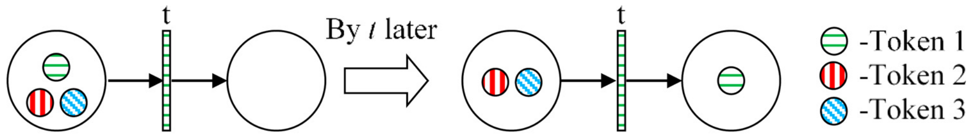
Despite their flexibility, conventional PNs meet difficulty when modelling complex systems or systems that are designed to deliver complex tasks and missions [30]. To address this issue, CPN was proposed [25]. In contrast to conventional PNs, the CPN is more informative as the tokens in the CPN are characterised by different colours, which represent different identities or other information. In order to enhance the flexibility of CPN, guard functions are designed to further constrain the enabling conditions of transitions. A guard function is a Boolean expression associated with a transition. A transition is only enabled if all the requirements of the general enabling rules are met and the defined guard function associated with the transition is true. To ease understanding, two CPN transitions are illustrated in Fig. 2.

To ensure that the tokens are distinguishable, the different colours are also associated with different patterns as defined in the key given in Fig. 2. In Fig. 2(a), the guard function is represented by a rectangular filled in green, which indicates only green tokens are allowed to transit through it. Hence after time delay t, the green token will be taken out of the input place and put into the output place. Besides, as shown in Fig. 2(b), only tokens with the same colour can enable the transition. No further firing of the transition will occur in Fig. 2(b) as there are not enough tokens of the same colour to activate the transition.

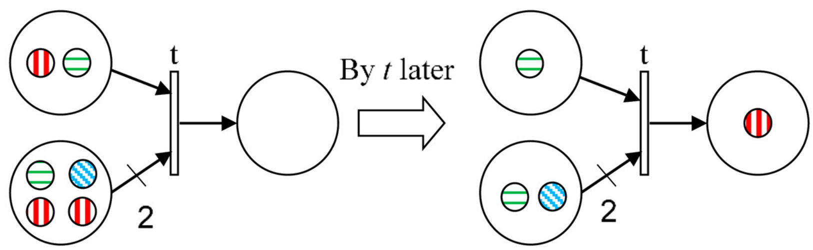
3. System layout

It is known that modelling a large scale AGV system is difficult and computationally expensive [31]. The modelling will be started with the identification of a system layout. The layout will be composed of important elements of an AGV system, such as stations for different purposes, paths connecting stations and AGVs running on the paths. An example of the layout of the AGV system, consisting of 9 stations (S1 to S9) and 12 bidirectional paths, is shown in Fig. 3. For simplicity, it is assumed in Fig. 3 that all stations have the same size and all paths have the same length. Such settings, characterised by repeated transportation patterns, are commonly adopted in the practical design of many large automated warehouses for ease of management [2].

A typical mission of the single-load AGVs is composed of five phases, i.e. (1) mission allocation and route optimisation, (2) dispatch



(a) CPN with a coloured transition.



(b) CPN with a normal transition.

Fig. 2. The CPN transitions.

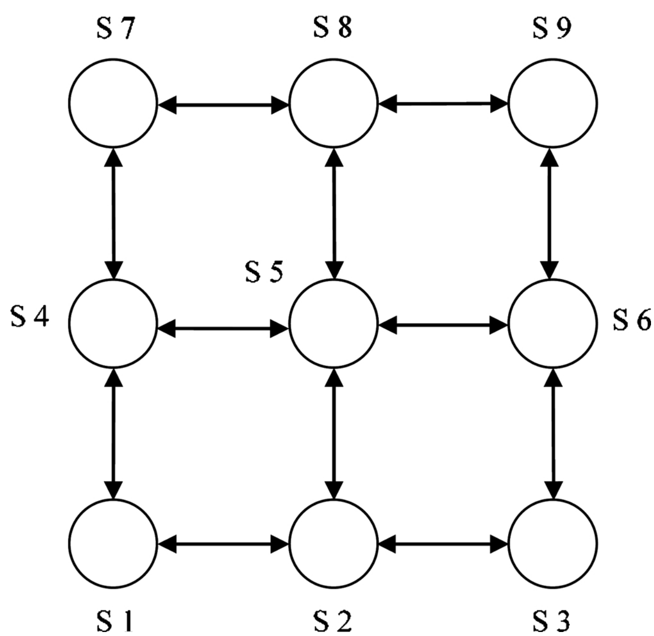


Fig. 3. System layout.

Table 1
Assumed phase durations.

Phases	Phase Duration (hour)
Phase 1: Mission allocation & route optimisation	0.005
Phase 3: Collect one item	0.02
Phase 5: Unloading	0.02

to the targeted pick-up station, (3) collection of single item, (4) travelling to the corresponding unloading station, and (5) unloading. Usually, the AGVs are not required to return to the base after unloading, so as they can start their next mission directly. In this work, values of the time durations for completing phases 1, 3 and 5 have been assumed for demonstration purposes and are given in Table 1. The time that is taken to complete phases 2 and 4 is dependent on the distance that the AGVs need to travel to deliver the assigned tasks.

In the application of multi-load AGVs, the mission is similar but there are usually multiple pickup and unloading stations. As multi-load AGVs possess a more powerful load capacity than single-load AGVs, the multi-load AGVs are expected to pick up items from multiple stations and transport them to prescribed unloading stations. In Fig. 3, station S1 is defined as the base of the AGVs, where the AGVs are stored and charged. Therefore, pickup and unloading activities will not happen at station S1. To guarantee the flexibility of the system, thereby making the modelling results more realistic and valuable for optimising and managing future AGV systems, the pickup and unloading stations in each mission will be randomly selected from stations S2-S9. Since the 12 bidirectional paths that connect the stations are assumed to have the same length, the AGVs will take the same time when travelling on any of them. In this paper, it is assumed to be 0.1 h. However, the layout of the system, speed of AGVs, and time taken by AGVs travelling on each path may be different case by case. They should be reset based on the actual situation and requirements of the system of interest. In addition, paths in the system layout of many real AGV systems are assumed to be bidirectional, but their width only allows one AGV to go through in order to save space and reduce the use of magnetic tape for navigation. As a consequence, all AGVs travelling on the same path should move in the same direction, otherwise, a deadlock will happen. Finally, the capacity of the 9 stations in the layout is assumed unlimited, so that the occupation of stations by AGVs will not cause blockage to other AGVs entering the same station.

4. The modelling of AGV systems

As this study is going to investigate the impact of load capacity of multi-load AGVs on system performance, all AGVs in the systems are assumed reliable so that the influence of AGVs' reliability, which has been investigated by Yan, Dunnett, and Jackson [24], can be neglected. In addition, if there are multiple AGVs in the systems, they should be distinguishable.

To simulate the operation of AGVs in different scenarios, three different types of CPN models, namely Route Petri Nets (RPNs), Master Petri Nets (MPNs), and Conflict Detection and Avoidance Petri Nets (DAPNs), will be developed. Among them, the RPNs are used to describe the routes that AGVs will travel along; MPNs are used to govern phase changes in missions; DAPNs are used to detect and avoid conflicts occurring in the AGV systems.

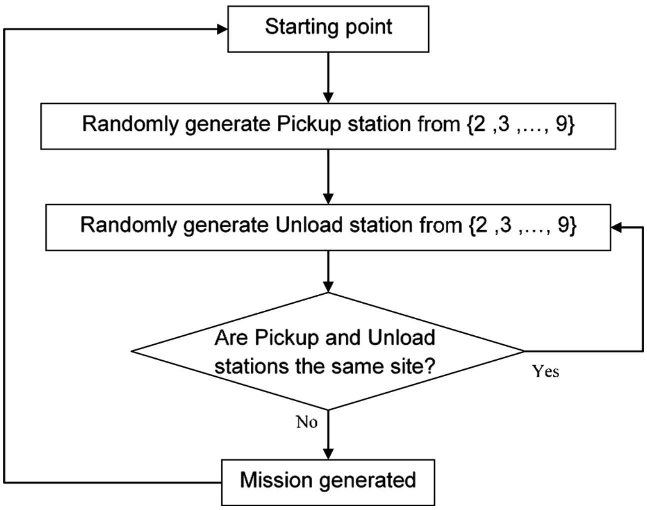


Fig. 4. Mission generation for a single-load AGV.

4.1. Mission generation

In a typical AGV system, there are three basic types of stations, i.e. starting point, pickup station, and unloading station. In the system layout in Fig. 3, the base of the AGVs, i.e. station S1, is also defined as the starting point of the initial mission for all AGVs. The starting points of the AGVs in subsequent missions will be their endpoints in the last mission. In the process of modelling, the mission specification is generated randomly, every station in the system layout except S1 will have an equal chance to be selected as either a pickup or unloading station. Since the multi-load AGV system will be compared with the corresponding single-load AGV systems that are used to complete the same workload, the mission generation algorithm dedicated to single-load AGV systems is also developed in this research. In this case, relevant rules need to be defined in advance to avoid that the same station is used as both pickup and unloading stations in the same mission. The algorithm is shown in Fig. 4.

Assume that the mission of a multi-load AGV consists of several tasks that need the vehicle to deliver continuously, the mission generation process for a multi-load AGV system is shown in Fig. 5, where 'n' indicates the load capacity of the multi-load AGV. In the process of modelling, the same weight and the same packed size of each item are assumed, which is commonly observed in warehouses dealing with items or products with the same standard.

From Fig. 5, it is seen that, as compared to the mission generation process of single-load AGVs, the random generation process of multi-load AGVs has to be repeated in the process of completing the generation of the mission. Each subtask is designed to deliver a particular item from one station to another. Again, the pickup station and the paired unloading station should be different in the same subtask. The repeating process will be ended until the load capacity of the multi-load AGV is fully used by the planned tasks, i.e. when $i = n$. It is assumed that there are always a sufficient number of subtasks queued in the system which is often the case for large warehouses.

4.2. Route optimisation and Route Petri Nets (RPNs)

Once the target stations in an AGV mission are known, routes to these target stations need to be optimised in order to obtain the shortest distance that AGVs need to travel to deliver all tasks. To facilitate route optimisation, a coordinate system is defined and is shown in Fig. 6.

The priority of the direction of movement of the AGVs between stations is defined by the following 5 rules:

Rule 1 – Prior to reaching a station that has the same x-coordinate as that of the target station, AGVs move in a horizontal direction.

Rule 2 – Once AGVs reach a station that has the same x-coordinate as that of the target station, its next movement will be in the vertical direction.

Rule 3 – When the x-coordinate of the target station is larger than that of the current station visited by the AGV, it will take the path on the right as the priority route. Otherwise, it will take the path on the left.

Rule 4 – When the y-coordinate of the target station is larger than that of the current station visited by the AGV, it will take the top vertical path as the priority route. Otherwise, it will take the bottom vertical path.

Rule 5 – When the AGV reaches a station that has the same x and y-coordinates as the defined x and y-coordinate values of the target station, it is assumed that the AGV has reached the target station in the present mission.

It is worth noting that these rules can be modified according to the actual needs of different layouts. If the paths are staggered or have different lengths, a route searching algorithm should be employed. However, implementation of a route searching algorithm will reduce simulation efficiency and hence should only be adopted when necessary. This is another reason for adopting the same length paths in the layout in this work, so that the computation time of the control center can be reduced. To demonstrate these rules, an example is shown. In the example, the AGV starts from station S1, and finally reaches target station S9. The optimal route that will be taken by the AGV is shown in Fig. 7.

Since different AGVs will be assigned different tasks, their target stations may be different from each other. Therefore, the RPN for every AGV must be generated separately. The RPN for the example described in Fig. 7 is illustrated in Fig. 8.

In the application of multi-load AGVs, it is particularly important to optimise the visiting order of the target stations, as different visiting order arrangements may lead to significantly different travelling distances in completing the same mission. For example, if a three-load AGV is expected to visit 3 target stations S4, S7 and S9, this can be achieved by a few route arrangements. For example, the AGV can travel along route S1→ S4 → S5→ S6→ S9→ S8→ S7, passing a total of 7 stations, or it can travel along route S1→ S4 → S7→ S8→ S9, passing only 5 stations. Therefore, visiting orders of the target stations should be optimised to ensure the shortest journey time and distance. To facilitate the optimisation of the visiting order of the target stations, a specific matrix is created to describe the shortest path distances corresponding to different pairs of pick up and unloading stations. In the example considered, the matrix is

$$
\begin{array}{cc}
 & \begin{array}{ccccccccc} \mathbf{S1} & \mathbf{S2} & \mathbf{S3} & \mathbf{S4} & \mathbf{S5} & \mathbf{S6} & \mathbf{S7} & \mathbf{S8} & \mathbf{S9} \end{array} \\
\begin{array}{c} \mathbf{S1} \\ \mathbf{S2} \\ \mathbf{S3} \\ \mathbf{S4} \\ \mathbf{S5} \\ \mathbf{S6} \\ \mathbf{S7} \\ \mathbf{S8} \\ \mathbf{S9} \end{array} &
\begin{bmatrix}
0 & 1 & 2 & 1 & 2 & 3 & 2 & 3 & 4 \\
1 & 0 & 1 & 2 & 1 & 2 & 3 & 2 & 3 \\
2 & 1 & 0 & 3 & 2 & 1 & 4 & 3 & 2 \\
1 & 2 & 3 & 0 & 1 & 2 & 1 & 2 & 3 \\
2 & 1 & 2 & 1 & 0 & 1 & 2 & 1 & 2 \\
3 & 2 & 1 & 2 & 1 & 0 & 3 & 2 & 1 \\
2 & 3 & 4 & 1 & 2 & 3 & 0 & 1 & 2 \\
3 & 2 & 3 & 2 & 1 & 2 & 1 & 0 & 1 \\
4 & 3 & 2 & 3 & 2 & 1 & 2 & 1 & 0
\end{bmatrix}
\end{array}
\tag{1}
$$

where the $[i, j]$-th element is the minimum number of path segments between station Si and station Sj. For example, any shortest route between S1 and S9 involves 4 path segments. The larger the number, the longer the distance between the stations. Consider the example mentioned earlier, in which a three-load AGV is expected to visit 3 target stations S4, S7 and S9. If S1 is the start point, the shortest distances to S4, S7 and S9 are 1, 2 and 4 path segments respectively. This suggests that the first station that the AGV should visit is S4. Then from S4, the shortest distances to S7 and S9 are 1 and 3, respectively. Therefore, the AGV should visit S7 and finally S9.

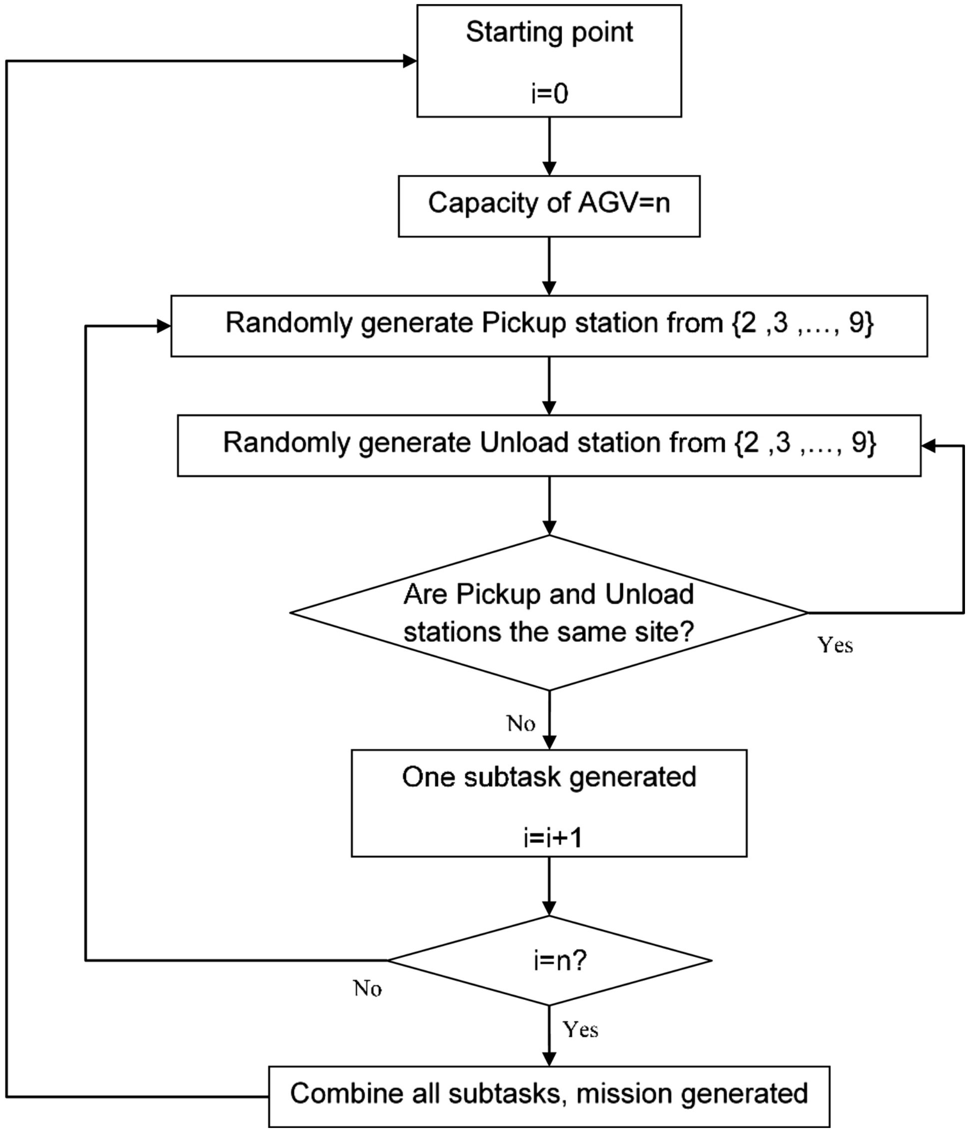


Fig. 5. Mission generation for a multi-load AGV.

4.3. Master Petri Nets (MPNs)

The MPN model is developed to govern the change of phases, as described in Section 3, from the beginning of the mission, Phase 1, to the successful completion of the whole mission, at the end of Phase 5. For example, the structure of the MPN for 3 single-load AGVs is illustrated in Fig. 9.

Once the travelling route of the AGVs is determined through optimisation, the MPN will be developed to define the phase changes for the operation of different AGVs. It is worth noting that in the MPN, AGVs are represented by different coloured tokens. Only the same coloured tokens in the places of both the RPN and MPN can enable transitions. In this way, the movement of AGVs and their working phases can be successfully correlated.

In the scenario of using single-load AGVs as shown in Fig. 9, there are 3 single-load AGVs in the system. They are initially at place 'Phase 1'. Once an AGV in 'Phase 1' is assigned a mission, the corresponding token will move from 'Phase 1' to 'Phase 2'. After the token is transferred to the 'Phase 2' place, the transitions between 'Phase 2' and the places in RPNs will be enabled. As the arcs connecting the transitions and the place 'Phase 2' are double-headed, a token of each colour will be placed back into place 'Phase 2' as well as the corresponding RPN's. The arc with a solid circle end is known as an 'inhibitor arc', which disables the connected transition once a token is present in the corresponding RPN. Following the movement of tokens in the corresponding RPNs, the transition between 'Phase 2' and 'Phase 3' is enabled and the corresponding token will move into place 'Phase 3' initiating travel to the unloading station. Since the route of every AGV is different, AGVs will take different amounts of time in their RPNs. Similar movements of AGVs and their transitions will continue until the mission is completed. Then, the AGVs will be assigned new missions and they will start from 'Phase 1' again.

In the scenario of using multi-load AGVs, as there are multiple pairs of pickup and unloading stations, the MPN developed in this scenario will be different from that shown in Fig. 9. In the operation of multi-load AGVs, after picking up items the AGV will not necessarily go directly to the unloading station for those items as a multi-load AGV is usually asked to deliver multiple items that are generated and packed together, as shown in Fig. 5. For example, on the way to the target unloading station, if the multi-load AGV passes a station and it is carrying some items that need to be unloaded there, the multi-load AGV will be expected to unload those items at that station before continuing travel to the target station. To ease understanding, an example of the MPN for a two-load AGV is shown in Fig. 10, where it is assumed that

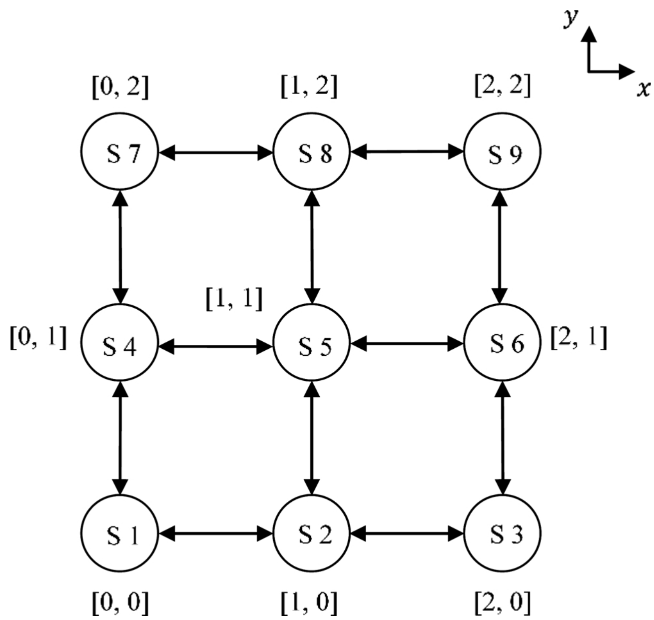


Fig. 6. The coordinate system for route optimisation.

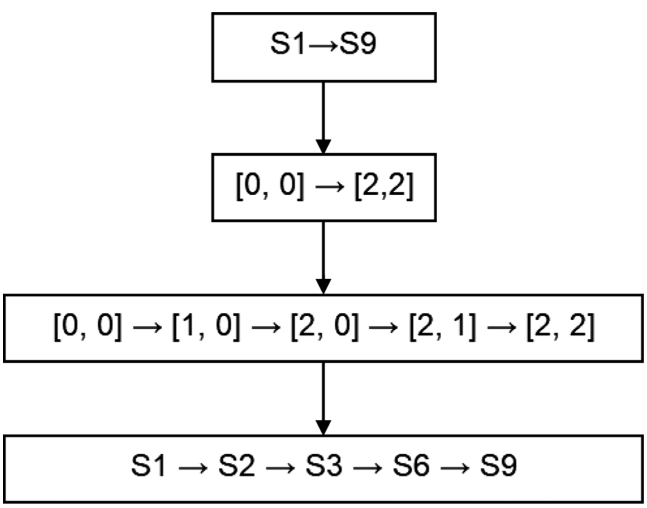


Fig. 7. The optimal route from station S1 to station S9.

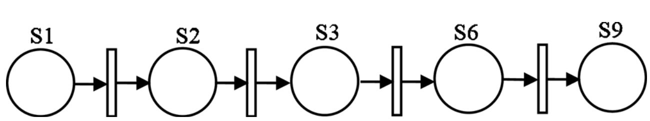


Fig. 8. The RPN for the example in Fig. 7.

the AGV is requested to deliver two tasks. The first is to transfer item A from station S2 to station S8, and the second is to transfer item B from station S6 to station S8.

From Fig. 10, it is seen that the AGV will visit all pickup stations first. So, the AGV will pass through the paths in the order of S1→S2→S3→S6. It is worth noting that the delay time of the transition for pickup or unloading action is always shorter than the delay time of the transition for moving to the next station. Such settings will guarantee that the vehicle will always conduct pickup or unloading action first before moving to the next station. The token in S6 will be transferred to 'pick up item B' place while a new token will be produced in S6 simultaneously due to the double-headed arrow. In the meantime, the transition connecting S6 and 'pick up item B' place will be disabled by the newly produced token in 'pick up item B' place due to the inhibitor arc, so that the newly produced token in S6 will be transferred to S5. Then, the AGV will travel to the unloading station. Once both items A and B are unloaded at the target unloading station S8, the mission is successfully completed. It should be noted that the PNs for pickup and unloading actions are temporary, they will be removed once the corresponding actions are completed. Therefore, the RPN and MPN of the multi-load AGV are not separable.

4.4. Petri Nets for describing conflict detection and avoidance (DAPNs)

It is well known that traffic conflict is inevitable particularly in the transportation of multiple AGV systems [2,32]. To facilitate the detection and avoidance of potential conflicts, the paths that all AGVs are travelling on must be identified and compared. To ease understanding of this method, an example is given in Fig. 11.

In Fig.11(b), if 'AGV 1' represented by the red token is travelling on 'path 2-1' while 'AGV 2' represented by a green token is going to travel on 'path 1-2', a conflict will occur. Then, the corresponding 'conflicts?' transition will be enabled and judge whether there is a conflict or not. It should be noted that this kind of instant transition can be enabled by tokens with different colours and will not change the colour property of the tokens being transferred. The token in the 'Next path' place will be transferred and produced simultaneously through the 'Conflict' place. This process will be repeated. It is worth noting that the token in the 'Conflict' place always has the same colour as the token in the 'Next path' place. Since there is a small-time delay, δt, for transferring the token to the place indicating the path being occupied, the token cannot move to its next place in the RPN. This means 'AGV 2' will have to wait in station S1 until 'AGV 1' reaches S1. Once 'AGV 1' reaches S1, the token in the place indicating 'path 2-1' will be transferred to the RPN of 'AGV 1' so that the token in 'Next path' will be transferred to the 'path 1-2'. Thus, the token in the 'path 1-2' place can enable the corresponding transition in the RPN of 'AGV 2'. Therefore, the DAPNs can control the movement of tokens in the RPNs.

5. Multi-load AGV simulation

With the aid of the three different types of CPN models developed in Section 4, a CPN-based model is developed in Python to simulate the mission, routing, and conflict detection and avoidance that often occurs in AGV systems. The MPNs govern the phase change of AGVs' missions, with the actions including route optimisation, pickup and drop off items being performed. The RPNs are automatically generated based on the algorithms of mission generation and route optimisation. It receives information from the MPNs if the AGVs are going to move. On the other hand, the RPNs can generate information about the next paths that the AGVs are going to travel on. These paths are compared in DAPNs to detect potential conflicts. The overall link between the three CPNs is illustrated in Fig. 12.

5.1. Model development and verification

To fulfill the purpose of this research, the influence of the load capacity of multi-load AGVs on system performance is investigated and compared with the corresponding single-load AGV systems that complete the same workload. The simulation procedure is implemented by the following steps:

Step 1: Initialise the model through:

(1) defining the values of the timed transitions. These times are based on the phase durations listed in Table 1 and the time required to complete the trip on every single path (i.e. 0.1 h as mentioned in Section 3).
(2) placing coloured tokens that represent different AGVs in Phase 1.

Step 2: Identify and switch the enabled transition with the minimum

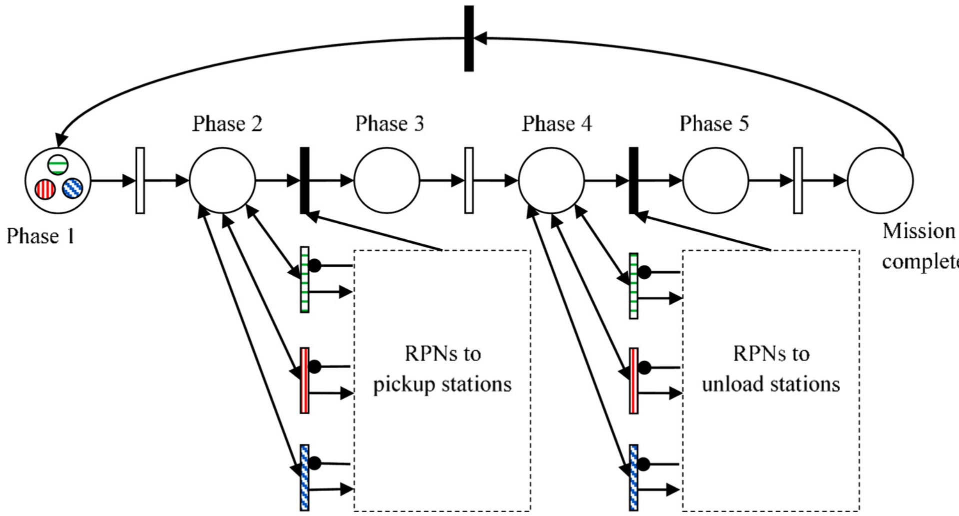


Fig. 9. The MPN for 3 single-load AGVs.

switching time in the whole model.

Step 3: Check all immediate transitions such as conflict checks. If any are found enabled, fire it.

Step 4: Repeat Step 3 until no more immediate transition is enabled.

Step 5: Check the condition of 'Have *n* number of missions (or subtasks for a multi-load AGV) been completed', if 'Yes', start next simulation; if 'No', repeat Steps 2–4.

Step 6: Iterate the above simulation until the predefined iteration time is reached.

In order to investigate the convergence performance of the developed model, the total operating time that is taken for completing 15,000 missions by three single-load AGVs is calculated. Each AGV is allocated 5,000 missions. The total operating time is defined as the sum of the operating time of all AGVs in the system. The calculation results are shown in Fig. 13.

From Fig. 13, it is found that the oscillation of the total operating time taken by all AGVs decays quickly with the increase of the number of simulations, and converges to about 6,193 h after the number of simulations is over 1,500.

5.2. Performance prediction of AGV systems

To facilitate comparison, the total operating time that is taken respectively by 1–10 single-load AGVs to complete 15,000 missions in the layout in Fig. 3 was calculated. It is assumed that there is no time gap between the tasks assigned to every AGV. The calculation results are listed in Table 2. The efficiency loss η can be calculated by using the following equations:

$$OT_i = T_i \times i \tag{2}$$

$$\eta = (OT_i - OT_1)/OT_1 \times 100\% \tag{3}$$

where i is the number of single-load AGVs in the system, T_i is the time taken to complete 15,000 missions by the system with i single-load AGVs, and OT_i is the total operating time of i single-load AGVs.

From Table 2, it is found that the more single-load AGVs are used, the less time will be taken by the system to complete the missions (see the results listed in the second column). However, the total operating time of all AGVs listed in the third column increases with the increase in the number of AGVs, due to the increased traffic conflicts as shown in the fourth column. This implies that the efficiency of the AGV system will decrease when more single-load AGVs are employed.

Subsequently, an AGV system that has the same system layout but contains only one multi-load AGV is considered to investigate the influence of the load capacity of multi-load AGVs on the performance of the system. As the visiting order of the stations can significantly affect the performance of the multi-load AGV system, the following 4

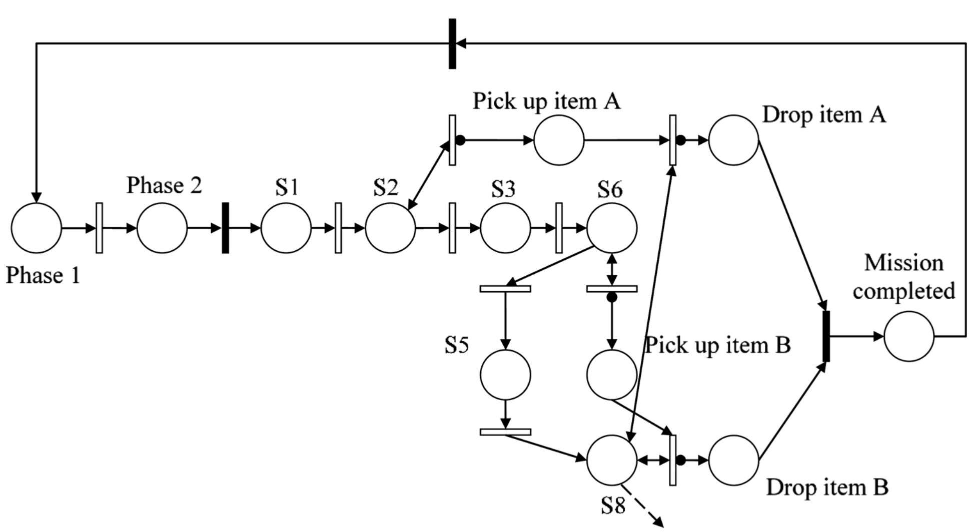
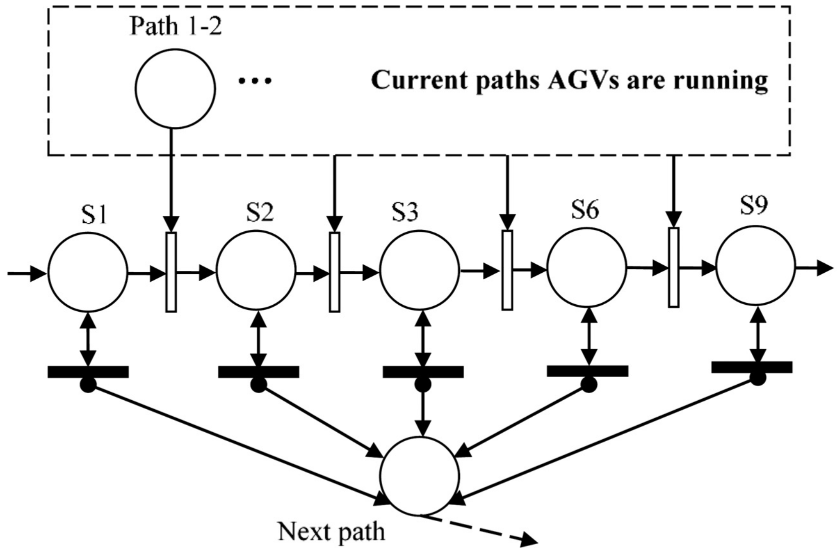
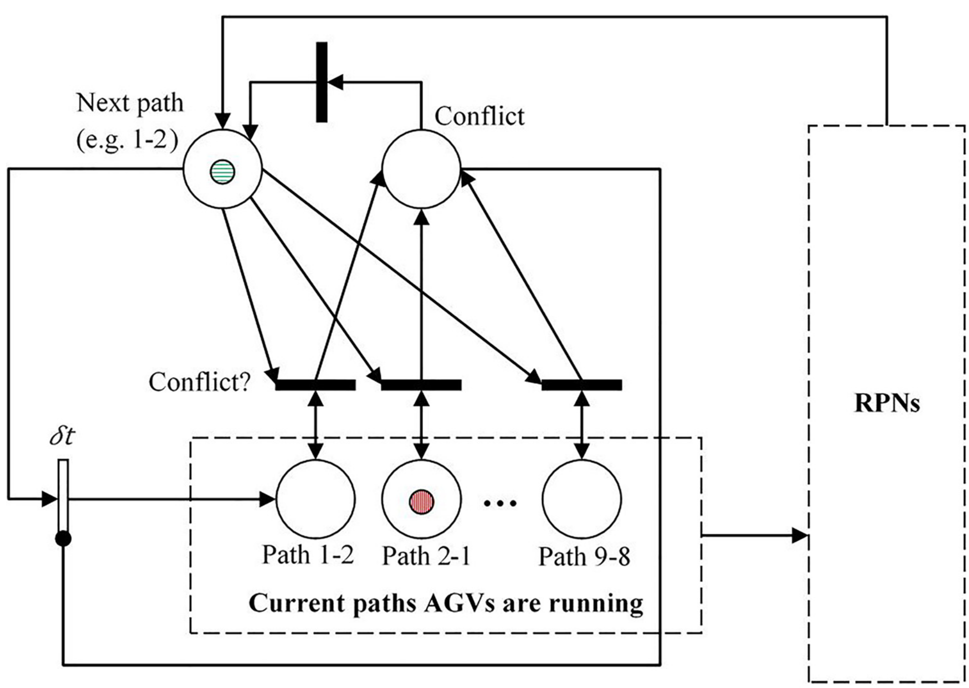


Fig. 10. The MPN for a two-load AGV.



(a) Example for identifying next path using RPN.



(b) Avoid conflict.

Fig. 11. The example of DAPNs.

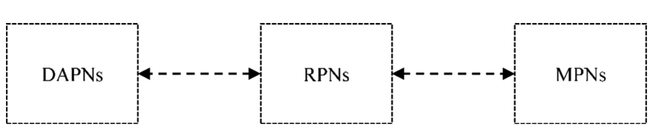


Fig. 12. CPN model integration.

scenarios with different station visiting order arrangments are considered:

1 First Come First Served (FCFS) scenario – as described in [33], the order of target stations is not optimised. The route the vehicle travels is based on the order of the subtasks received;
2 Optimise the visiting order of the target stations – the order of the target stations is optimised to ensure the shortest distance of the journey;
3 If all the missions were known in advance, the missions sharing the same stations can be grouped together for completion. Given the visiting order of the target stations is optimised but the unloading station is fixed in each mission – in other words, all the subtasks in a multi-load AGV mission have the same assigned unloading station. Hence, a vehicle can pick up all the items first and then travel to the unloading stations.
4 Assemble the subtasks that have both the same loading and same

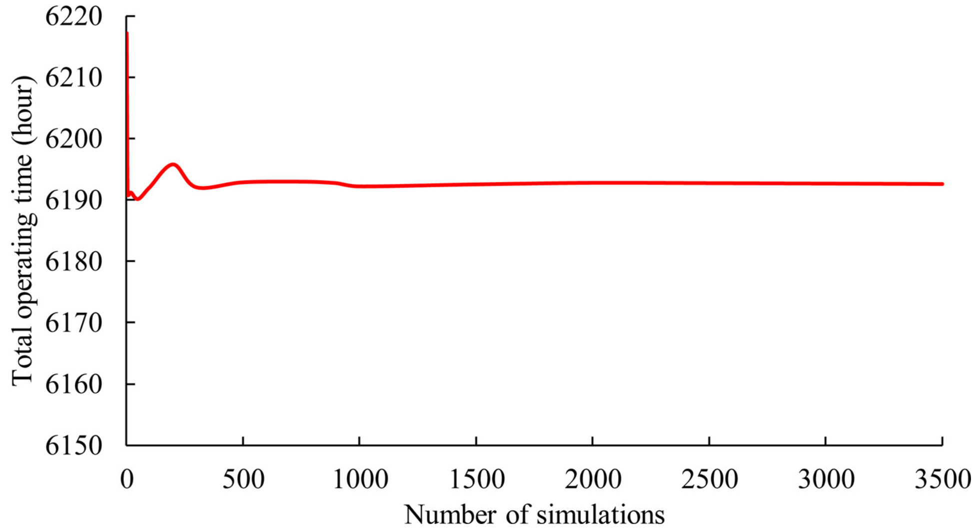


Fig. 13. Convergence performance of the developed model.

Table 2
Total operating time that is taken by different single-load AGV systems.

Number of single-load AGVs	Time for completing 15,000 missions T_i (hour)	Total operating time of all single-load AGVs OT_i (hour)	loss in operational efficiency η	Number of conflicts in the process
1	6099.6060	6099.6060	0.00 %	0
2	3075.6296	6151.2593	0.85 %	1755.9603
3	2064.2229	6192.6688	1.53 %	3396.8913
4	1556.8209	6227.2837	2.09 %	4922.8177
5	1251.0290	6255.1448	2.55 %	6340.2453
6	1046.6469	6279.8813	2.96 %	7666.1243
7	900.06983	6300.4888	3.29 %	8885.5333
8	789.79646	6318.3717	3.59 %	10064.104
9	703.74053	6333.6648	3.84 %	11153.409
10	634.74161	6347.4161	4.06 %	12183.783

unloading stations to one mission to fully utilise the load capacity of the multi-load AGV. As a consequence, there will be only one pair of pickup and unloading stations in each mission.

It is assumed that the multi-load AGV always travels at a constant speed and its load capacity varies from 1 to 10 items. The corresponding total operating time of the system for completing 15,000 missions in the above 4 scenarios is calculated and the results are graphically illustrated in Fig. 14.

From Fig. 14, it is found that

(1) In all of the four scenarios, the operating time shows a gradually decreasing tendency with the increase of the load capacity of the multi-load AGV. This proves that the operation efficiency of the AGV system can be improved by increasing the load capacity of the multi-load AGVs;
(2) The curve representing the operating time in scenario 2 is always below the curve representing the operating time in scenario 1 under all AGV load capacity conditions other than when the load capacity is '1'. This suggests that the performance of the multi-load AGV system can be improved through optimising the visiting order of the target stations;
(3) Further observation of Fig. 14 has shown that based on the optimised visiting order of the target stations, system efficiency can be further improved if the pickup and unloading stations in a mission are fixed, although the feasibility of this should be investigated in the practical design of an AGV system.

To demonstrate the importance of the optimisation of station visiting order on system efficiency, the improvement of system efficiency in the aforementioned 4 scenarios is calculated by:

$$\Delta E = (T_{FCFS} - T)/T_{FCFS} \times 100\% \quad (4)$$

where ΔE indicates the improvement of system efficiency, T_{FCFS} is the operating time of the multi-load AGV in scenario 1, T is the operating time of the multi-load AGV in the other scenarios considered. The calculation results of ΔE are graphically shown in Fig. 15.

From Fig. 15, it is seen that efficiencies increase rapidly at the beginning when the load capacity of the AGV increases from 1 to 3, since which the increasing tendencies slow down and finally converge to constant values after the load capacity of the AGV is over 6. Hence after exceeding a capacity of '6' in this example, the efficiency improvement via increasing the load capacity of the multi-load AGV will be limited. This implies that there must be an 'optimal load capacity' for the multi-load AGV. This knowledge will be very important for scaling up the future design of AGV systems. In addition, Fig. 15 shows that the maximum efficiency improvement happens in Scenario 4. This suggests that the shorter the route in a mission, the more the efficiency will be improved through increasing the load capacity of the AGV.

The operating time of the single-load and multi-load AGV systems is also compared to further demonstrate the impact of the load capacity of multi-load AGVs and their potential advantages over using multiple single-load AGVs. In order to present the results in a readable format only results obtained in the application of the multi-load AGV in the most inefficient scenario (Scenario 1) and the most efficient scenario (Scenario 4) are considered for comparison. The comparison results are shown in Fig. 16.

From Fig. 16, it is seen that, except when the load capacity is '1', the curves representing 'Time taken by the multi-load AGV to complete missions' are always above the corresponding 'Time taken by single-load AGVs to complete missions'. This indicates that in contrast to using a multi-load AGV, using multiple single-load AGVs can reduce the time required to complete missions. However, the solid line that represents 'Total operating time of all single-load AGVs' is found always above the curves obtained when using the multi-load AGV other than when the load capacity is '1'. This suggests that although using multiple single-load AGVs can reduce mission completion time, it leads to a much longer total system operating time than when using a multi-load AGV. Moreover, from Fig. 16 it can be found that when the multi-load AGV is operated in Scenario 1, the efficiency of two single-load AGVs can be reached when the load capacity of the multi-load AGV is 8. By contrast, when the multi-load AGV is operated in Scenario 4, the efficiency of

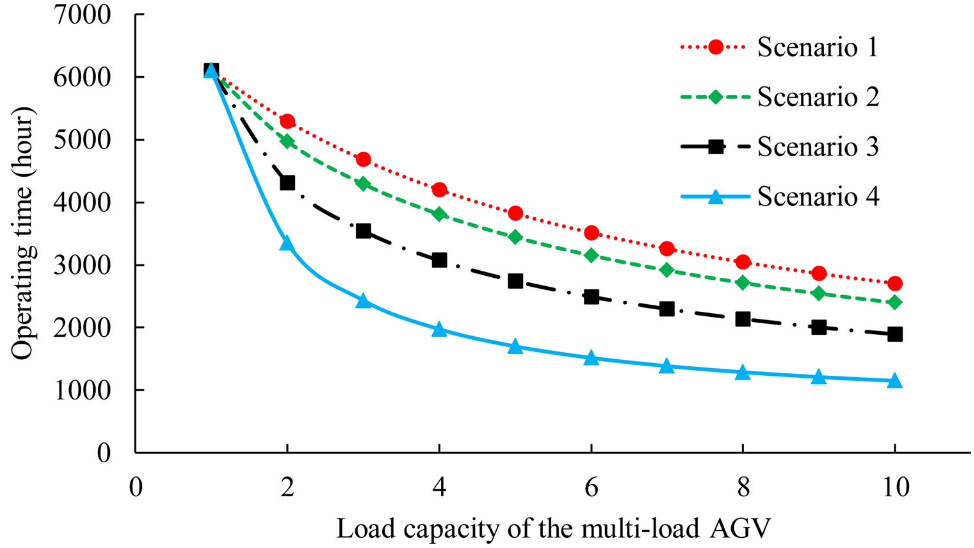


Fig. 14. The tendencies of operating time versus the capacity of multi-load AGV.

two single-load AGVs can be nearly reached when the load capacity of the multi-load AGV is 2. This proves that multi-load AGVs can improve system performance, but their contribution to performance improvement is highly dependent on the appropriate use of them.

6. Conclusions

In order to further explore the advantages of using multi-load AGVs, three novel CPN models have been developed in this paper to simulate the operation of a multi-load AGV. Through investigating the impact of the load capacity of multi-load AGVs on system performance and comparing it with the corresponding single-load AGV systems that can complete the same workload, the following conclusions are drawn:

(1) As opposed to conventional mathematical programming methods, the CPN models proposed in this paper do provide a powerful tool to simulate the operation of the systems that have multiple AGVs;
(2) Increasing the number of single-load AGVs in a system can reduce mission completion time. However, it will lead to an increase in the total operating time of all single-load AGVs. Therefore, the efficiency of AGV systems will be decreased when more single-load AGVs are running in the system due to the increased traffic conflicts;
(3) Compared to using multiple single-load AGVs, the use of a single multi-load AGV does lead to a longer mission completion time. However, the total operating time when using a multi-load AGV is much lower than when using multiple single-load AGVs. This suggests that a system with flexible loading and unloading points will be more efficient if a multi-load AGV is used;
(4) An appropriate dispatching rule is critical in the application of multi-load AGVs. The optimisation of it can increase the contribution of the multi-load AGV to system efficiency. In this research, it is found that the best arrangement is that adopted in Scenario 4, but this is only applicable when the tasks are known in advance;
(5) The research has shown that after the load capacity of the multi-load AGV exceeds a certain value, system efficiency will no longer

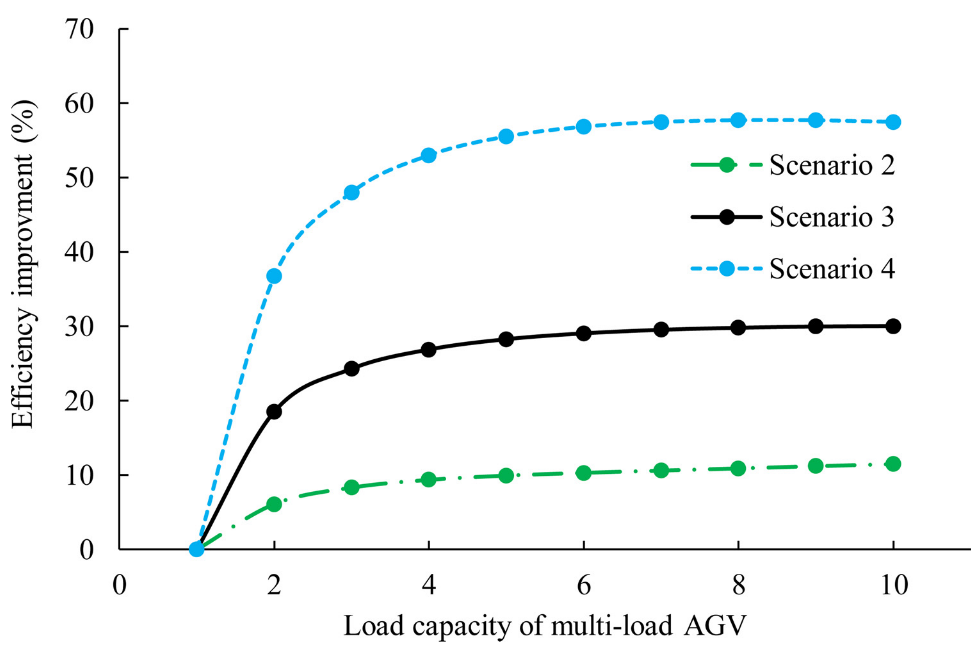


Fig. 15. Efficiency improvement in different multi-load AGV scenarios.

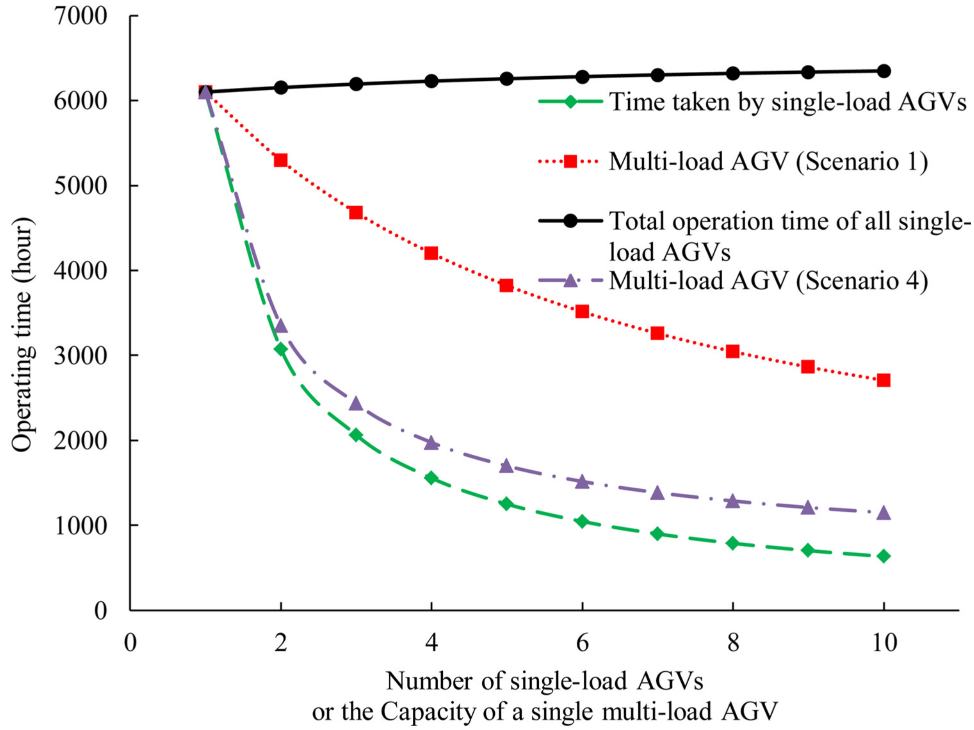


Fig. 16. Performance comparison of multi-load and single-load AGVs.

be significantly improved through further increasing the load capacity of the AGV. This means that there is an optimal load capacity of the multi-load AGV for a particular system. This knowledge is of great significance for scaling up of the future AGV systems.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work reported above was supported by EPSRC [grant number EP/K014137/1].

References

[1] Le-Anh T, De Koster MBM. A review of design and control of automated guided vehicle systems. Eur J Oper Res 2006;171:1–23. https://doi.org/10.1016/j.ejor.2005.01.036.

[2] Vis IFA. Survey of research in the design and control of automated guided vehicle systems. Eur J Oper Res 2006;170:677–709. https://doi.org/10.1016/j.ejor.2004.09.020.

[3] Li ZW, Wu NQ, Zhou MC. Deadlock control of automated manufacturing systems based on Petri Nets-A literature review. IEEE Trans Syst Man Cybern C Appl Rev 2012;42:437–62. https://doi.org/10.1109/TSMCC.2011.2160626.

[4] De Ryck M, Versteyhe M, Debrouwere F. Automated guided vehicle systems, state-of-the-art control algorithms and techniques. Int J Ind Manuf Syst Eng 2020;54:152–73. https://doi.org/10.1016/j.jmsy.2019.12.002.

[5] Wu NQ, Zhou MC. Modeling and deadlock control of automated guided vehicle systems. IEEE ASME Trans Mechatron 2004;9:50–7. https://doi.org/10.1109/TMECH.2004.823875.

[6] Bhattacharya R, Bandyopadhyay S. An improved strategy to solve shop deadlock problem based on the study on existing benchmark recovery strategies. Int J Adv Manuf Technol 2010;47:351–64. https://doi.org/10.1007/s00170-009-2200-y.

[7] Xin J, Negenborn RR, Corman F, Lodewijks G. Control of interacting machines in automated container terminals using a sequential planning approach for collision avoidance. Transp Res Part C Emerg Technol 2015;60:377–96. https://doi.org/10.1016/j.trc.2015.09.002.

[8] Luo J, Wu Y, Mendes AB. Modelling of integrated vehicle scheduling and container storage problems in unloading process at an automated container terminal. Comput Ind Eng 2016;94:32–44. https://doi.org/10.1016/j.cie.2016.01.010.

[9] Vivaldini K, Rocha LF, Martarelli NJ, Becker M, Moreira AP. Integrated tasks assignment and routing for the estimation of the optimal number of AGVS. Int J Adv Manuf Technol 2016;82:719–36. https://doi.org/10.1007/s00170-015-7343-4.

[10] De Ryck M, Versteyhe M, Shariatmadar K. Resource management in decentralized industrial Automated Guided Vehicle systems. Int J Ind Manuf Syst Eng 2020;54:204–14. https://doi.org/10.1016/j.jmsy.2019.11.003.

[11] Ozden M. A simulation study of multiple-load-carrying automated guided vehicles in a flexible manufacturing system. Int J Prod Res 1988;26:1353–66. https://doi.org/10.1080/00207548808947950.

[12] Liu FH, Hung PC. Real-time deadlock-free control strategy for single multi-load automated guided vehicle on a job shop manufacturing system. Int J Prod Res 2001;39:1323–42. https://doi.org/10.1080/00207540010005105.

[13] Ho Y-C, Chien S-H. A simulation study on the performance of task-determination rules and delivery-dispatching rules for multiple-load AGVs. Int J Prod Res 2006;44:4193–222. https://doi.org/10.1080/00207540500442401.

[14] Grunow M, Hans-otto G, Lehmann M. Dispatching multi-load AGVs in highly automated seaport container terminals. OR Spectr 2004;26:211–35. https://doi.org/10.1007/s00291-003-0147-1.

[15] Chawla VK, Chanda AK, Angra S. Simultaneous dispatching and scheduling of multi-load AGVs in FMS-A simulation study. Mater Today Proc 2018;5:25358–67. https://doi.org/10.1016/j.matpr.2018.10.339.

[16] Ho YC, Liu HC, Yih Y. A multiple-attribute method for concurrently solving the pickup-dispatching problem and the load-selection problem of multiple-load AGVs. J Manuf Syst 2012;31:288–300. https://doi.org/10.1016/j.jmsy.2012.03.002.

[17] Spedding TA, Sun GQ. Application of discrete event simulation to the activity based costing of manufacturing systems. Int J Prod Econ 1999;58:289–301. https://doi.org/10.1016/S0925-5273(98)00204-7.

[18] Duggan J. A comparison of Petri Net and system dynamics approaches for modelling dynamic feedback systems. 24th Int Conf Syst Dyn Soc 2006:1–22.

[19] Safarinejadian B. Discrete event simulation and Petri net modeling for reliability analysis. Int J Soft Comput Softw Eng 2012;2:25–36. https://doi.org/10.7321/jscse.v2.n5.3.

[20] Simon E, Oyekan J, Hutabarat W, Tiwari A, Turner CJ. Adapting Petri nets to DES: stochastic modelling of manufacturing systems. Int J Simul Model 2018;17:5–17. https://doi.org/10.2507/IJSIMM17(1)403.

[21] Hsieh S, Shih Y-J. Automated guided vehicle systems and their Petri-net properties. J Intell Manuf 1992;3:379–90. https://doi.org/10.1007/BF01473533.

[22] Nishi T, Tanaka Y. Petri net decomposition approach for dispatching and conflict-free routing of bidirectional automated guided vehicle systems. IEEE Trans Syst Man, Cybern Part A Syst Humans 2012;42:1230–43. https://doi.org/10.1109/TSMCA.2012.2183353.

[23] Yan R, Jackson LM, Dunnett SJ. Automated guided vehicle mission reliability modelling using a combined fault tree and Petri net approach. Int J Adv Manuf Technol 2017;92:1825–37. https://doi.org/10.1007/s00170-017-0175-7.

[24] Yan R, Dunnett SJ, Jackson LM. Novel methodology for optimising the design, operation and maintenance of a multi-AGV system. Reliab Eng Syst Saf 2018;178:130–9. https://doi.org/10.1016/j.ress.2018.06.003.

[25] Jensen K. Coloured petri nets: basic concepts, analysis methods and practical. Heidelberg, Germany: Springer; 1996.
[26] Wu N, Zeng W. Deadlock avoidance in an automated guidance vehicle system using a coloured Petri net model. Int J Prod Res 2002;40:223–38. https://doi.org/10.1080/00207540110073037.
[27] Murata T. Petri Nets: properties, analysis and applications. Proc IEEE 1989;77:541–80.
[28] Mejía G, Pereira J. Multiobjective scheduling algorithm for flexible manufacturing systems with Petri nets. J Manuf Syst 2020;54:272–84. https://doi.org/10.1016/j.jmsy.2020.01.003.
[29] Hu L, Liu Z, Hu W, Wang Y, Tan J, Wu F. Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network. J Manuf Syst 2020;55:1–14. https://doi.org/10.1016/j.jmsy.2020.02.004.
[30] David R, Hassane A. Petri nets for modeling of dynamic systems a survey. Automatica 1994;30:175–202. https://doi.org/10.1016/0005-1098(94)90024-8.
[31] Xin J, Negenborn RR, Lodewijks G. Energy-aware control for automated container terminals using integrated flow shop scheduling and optimal control. Transp Res Part C Emerg Technol 2014;44:214–30. https://doi.org/10.1016/j.trc.2014.03.014.
[32] Farling BE, Mosier CT, Mahmoodi F. Analysis of automated guided vehicle configurations in flexible manufacturing systems. Int J Prod Res 2001;39:4239–60. https://doi.org/10.1080/00207540110072957.
[33] Lee J, Tangjarukij M, Zhu Z. Load selection of automated guided vehicles in flexible manufacturing systems. Int J Prod Res 1996;34:3383–400. https://doi.org/10.1080/00207549608905096.