Improved Particle Swarm Approach for Dynamic Automated Guided Vehicles Dispatching

Radhia Zaghdoud[1], Marwa Amara[2]
Computer Science Department, Faculty of Science
Northern Border University, Arar 91431, Saudi Arabia[1, 2]
LARIA, National School of Computer Science
Manouba University, Tunis, Tunisia[1, 2]

Khaled Ghedira[3]
Université Centrale de Tunis
Tunis, 1002, Tunisia

***Abstract*—The automated guided vehicles dispatching is one of the important operations in containers terminal because it affects the loading/unloading process. This operation has become faster and more complex until the automation advent. Although this evolution, the environment has become dynamic and uncertain. This paper aims to propose an improved particle swarm approach for solving the bi-objective problem of automated guided vehicles dispatching and routing in a dynamic environment of containers terminal. The objectives are to minimize the total travel distance of all automated guided vehicles and maximize the workload balance between them. The application of particle swarm algorithm in its basic form, shows a premature convergence. To ameliorate this convergence, the authors proposed the application of a method to escape the worst particles from the local optimum. The new Hybrid Guided Particle Swarm approach consists of hybridization between Dijkstra algorithms and a Guided Particle Swarm Algorithm. The routing problem is solved with Dijkstra algorithm and the dispatching problem with guided particle swarm approach. As a first step, this approach has been applied in a static environment where the dispatching parameters and the routing parameters are fixed in advance. The second step consists of applying this approach in a dynamic environment where the number of containers associated with each automated guided vehicles can change, the shortest path and the container locations can also change during the algorithm execution. The numeric results in a static environment show a good Hybrid Guided Particle Swarm performance with a faster and more stable convergence, which surpasses previous approaches such as Hybrid Genetic Approach and the efficiency of its extension approach Dynamic Hybrid Guided Particle Swarm in a dynamic environment.**

Keywords*—Dispatching; automated guided vehicles; dynamic; containers; particle swarm; genetic algorithm

I. Introduction

The productivity of maritime transportation has significantly advanced with the advent of automation. Automated container terminals have become crucial intermediaries between the marine and land transportation systems. As the number of ships entering and exiting daily from terminals has greatly increased, the number of containers loaded by ships has become very large [1]. Following this revolution, many maritime ports worldwide have established automated equipment to manage the increase in container traffic. They also installed an automated control system to achieve an optimal performance. The equipment in ACT is classified into three principal types: quayside equipment, quay cranes (QCs) used to load or unload containers from or to the ship; landside equipment, yard cranes (YCs) employed for container loading or unloading in the yard storage depots; and intermediate zone equipment, automated guided vehicles (AGVs) used to transport containers from the two sides of the port. An AGV is a mobile robot that follows markers or wires on the floor or uses vision, magnets, or lasers for navigation. It is extensively employed in industry to transport goods from an origin location to a target location [2, 3, 4]. AGVs are widely used in manufacturing, medicine, and logistics industries. Although this equipment has accelerated the ACT operations, any working failure of any one of them may cause a late or partial blockage or global blockage of the whole system in the ACT. The most critical goal of an automated container terminal is to increase productivity by minimizing the berthing duration of ships. This objective can be accomplished by finishing the main ship loading/unloading operation at its scheduled time because any lateness can affect the synchronization of the entire system. This operation includes four types of sub-operations: (1) loading/unloading containers from/to the ship, (2) AGV dispatching and routing, and (3) loading/unloading containers to the yard storage zone. The container loading/unloading process began after ship berthing. The containers are unloaded from the ship by the quay cranes and are then transported by the AGVs to the landside of the port. They are then unloaded by yard cranes and stocked in the corresponding yard storage zone. Conversely, the ship-loading operation begins by assigning containers to AGVs for transfer to the quayside to be loaded by the quay cranes to the ship. These loading and unloading operations have become faster with the advent of container terminal automation and a high number of equipment in the port [5]. Nevertheless, with this important evolution of the container terminal, the risk of breakdown of any element in the system increases. This failure can have some consequences, such as the lateness of the loading/unloading operation; therefore, the berthing duration of the ship increases, which will indirectly disturb the productivity of the port. The latency of the ship berthing duration may cause the unavailability of container terminal equipment, which will delay the loading/unloading operation of all ships after this disruption. This paper studies the problem of AGVs dispatching in a static and dynamic environment of containers terminal. In the first step, the authors investigate to solve the static problem. They choose to optimize the total travel distance of all AGVs and the workload balance between them. By studying these two criteria; the total travel distance depends on AGV's path and the balance workload depends on AGV's autonomy. The first criteria were optimized by using an exact algorithm to search the shortest path for each AGV and the second criteria were optimized by maximizing the autonomy of each AGV. The workload balance makes the AGVs system more robust because the AGV working more tasks than other AGVs will lose its autonomy early. Because the AGV's autonomy is limited to its battery energy, the breakdown of AGVs can be frequent with absence of workload balance. In a static environment, this problem parameters are fixed in advance; the number of road network nodes, and the number of AGVs, and the number of containers. However, the real situation in maritime ports in completely different because the number of equipment is big, so the possibility of breakdown of any equipment is also big. A breakdown of any equipment can make a delay in the corresponding sub-process which can affect the loading /unloading operation. A new ship arriving to the port may haven't the necessary equipment of its unloading/loading operation available. An extension of the proposed approach in the first step, was developed to solve the problem of dispatching AGVs to containers in a dynamic environment. The remainder of this paper is organized as follows: the second Section presents a literature review of this problem. The problem description and mathematical formulation are presented in the third Section. In the fourth Section, we present the proposed approaches in a static environment. In the fifth section, the authors propose an extension of the proposed approach for the dynamic environment. The numerical study results are cited in the sixth section, and finally, a discussion of the results and conclusions is given in the seventh section.

II. Literature Review

A. Related Work

In ACTs, the operations can be grouped into two classes of processes: loading and unloading. The loading process consists of transferring containers from the yard location area to AGVs via yard cranes to be transported to the ship to be loaded by the quay cranes. The unloading process involves unloading containers from the ship using quay cranes, and transportation by AGVs to the corresponding storage locations in the yard. Many studies have focused on ACT operations. Most studies were concerned with global loading and unloading operations. They proposed simultaneous scheduling systems including QCs, AGVs, and YCs. Despite the importance of dispatching containers to AGVs and AGV routing in the unwinding of loading and unloading processes, few studies have independently focused on this problem. Several literature reviews were developed [7, 8, 9, 10] studying yard and quay side operations, examining independently studied problems, as well as combined problems. They also revised the literature on yard crane scheduling, transport vehicle dispatching and scheduling, quay crane assignment and scheduling problems for the yard, vehicle routing and traffic control, and storage location and space planning problems. [11] considered the global ACT system and proposed an integrated scheduling model for handling equipment coordination and AGV routing. The optimization goal was to minimize the makespan of the global process. The authors developed a Congestion Prevention Rule-based bi-level genetic algorithm (CPR-BGA) to solve the proposed model. [12] proposed a new method for optimizing the ASC and AGV scheduling and a collaborative AGV and ASC scheduling model in automatic terminal. The proposed model is designed based on a genetic algorithm (GA) and aims to minimize the AGV waiting time and ASC running time. The dispatching problem of AGVs to containers was studied by [6], where the AGVs scheduling was assimilated as a process of allocating AGVs to tasks, considering the cost and time of operations. The objectives chosen were makespan maximization and minimization of the number of AGVs, while considering the battery charge of the AGVs. A fuzzy GA, PSO optimization algorithm, and a hybrid GA-PSO were developed to optimize the proposed model. [13] proposed an approach named the modified memetic particle swarm optimization (MMPSO) algorithm based on PSO integrated with the memetic algorithm (MA). This approach is applied to generate the initial feasible solutions for scheduling multi-load AGVs to minimize travel and waiting time in manufacture system (FMS). [14] studied the problem of work transport organization and control. They proposed an approach based on a non-changeable path during travel and a fuzzy logic to order the set of stations requesting transport services. GA is applied to stations sequence optimization. [15] studied the problem of resource optimization in AGV-served FMS. They proposed a scheduling model integrating machines and AGVs. The objective function is the makespan of jobs from raw material storage to finished parts storage. [16] consider the problem of dispatching multiple-load AGVs in an FMS. A PDER rule based on pickup-or-delivery-in-route is proposed to address the task determination problem, which indicates whether the next task of an AGV partially loaded should be picking up a new job or dropping off a carried load. A workload-balancing (WLB) algorithm was developed to address the pickup-dispatching problem that determines which job should be assigned to an AGV. [17] investigated the multiple-AGV path planning. The authors proposes a GA approach with two innovations; a three-exchange crossover heuristic operators, used to produce better offspring and a double-path constraint for minimizing the total path distance of all AGVs and the single path distances of each AGV. [18] studied the autonomous driving system that uses dynamic path planning to avoid static and moving obstacles. To determine the optimal path, acceleration, and vehicle speed, the proposed method generated a set of path candidates. The optimum path selection is based on the total cost of static safety, comfortability, and dynamic safety, with the identification of acceleration and speed. [19] proposed a Q-learning method to find the AGVs shortest-time routes. To improve the selecting action policy for this method, the authors developed an improved anisotropic Q-learning routing algorithm with vehicle-waiting-time estimation. The performance of these methods was tested based on simulations. [20] considered the dynamic scheduling process to solve the AGV scheduling and planning problems. The authors proposes a two stage mixed integer model for AGVs cost transportation optimization with lay time constraint. They developed an approach based on heuristic, and DIK algorithm, and Q-learning algorithm for solving the proposed model. A strategy for avoidance conflict of AGVs was also proposed. [21] investigated the dynamic AGVs scheduling problem with AGVs and machines having specific speed. They proposed a biological intelligent approach (BIA) inspired by hormone regulation in endocrine system. The objectives were to minimize the makespan and maximize the shop floor work efficiency. To solve this problem in a static environment, many approaches have been proposed, and metaheuristics perform well for this type of problem [6]. The PSO algorithm is one of the best algorithms cited in the literature, although the disadvantage of its premature convergence. To the authors' best knowledge, the problem of dispatching containers to AGVs in container terminals has been studied in general, in an integrated manner with other dispatching problems, such as the dispatching of quay cranes and yard cranes. The weakness of the combination of this problem of dispatching with the other problems of dispatching in container terminals in the same system management can propagate any disruption from one phase to the following phase. For example, if there is a problem in quay crane dispatching, it will propagate to AGV dispatching. So, the resolution becomes more complex. The study of this problem separately can easily detect any disruption and facilitate its resolution. Many studies investigate the problem of AGV dispatching and routing in a static environment, but this approach is different of the real situation inn container terminals so its application will not be efficient. A scarce number of researchers are interested in this problem in a dynamic environment. All the previous studies don't combine the two criteria of travel distance and workload balance, although the combination of these two criteria can be more attached and applicable to the real situation. The hybridization of particle swarm with Dijkstra algorithm make a good guide to PSO approach to find the best solution and the re-initialization of worst particles parameters help these particles to avoid from local optimum. The PSO approach has the advantage of the fast convergence in comparison with genetic algorithm. In this paper, the authors propose a guided hybrid particle swarm algorithm (GHPSO) to solve this problem in a static environment. Because the accurate situation in the container terminal is not static, they propose an extension of this approach to apply in the dynamic environment.

B. Particle Swarm Optimisation

A particle swarm is a metaheuristic algorithm presented in 1995 by Kennedy and Eberhart, and it was developed under the inspiration of the behavior laws of bird blocks, fish scrolls, and human communities. To achieve the optimum solution, PSO starts from a group of random groups of solutions and then repeatedly searches. It has proven to be a highly efficient optimization algorithm in numerous studies and experiments [22]. As a metaheuristic, PSO does not guarantee that the optimal solution is obtained. The basic particle swarm optimization is described as follows:

Assuming N is the number of particles, the i^{th} particle position $i = (1,2, \dots, N)$ in dimension space d can be denoted as $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,d}]$, its velocity is defined as the moving distance between the particles in each iteration, and is denoted as $V = [v_{i,1}, v_{i,2}, \dots, v_{i,d}]$.

The objective function consists of determining the optimal position of the particle, and the local optimal particle position P_{best} in the t^{th} iteration is denoted as $P_i = [p_{i,1}, p_{i,2}, \dots, p_{i,d}]$. The global optimal position gbest in the t^{th} iteration is denoted as $P_g = [p_{g,1}, p_{g,2}, \dots, p_{g,d}]$. In the $(t+1)^{th}$ iteration, the flight velocity $V_{i,j}(t+1)$ of the i^{th} particle in the j dimensional space, $j = (1,2, \dots, d)$, and its position $X_{i,j}(t+1)$ can be derived from the following equations:

$$V_{i,j}(t+1) = W * V_{i,j}(t) + C_1 * R_1 * [P_{i,j} - X_{i,j}(t)] + C_2 * R_2 * [P_{g,j} - X_{i,j}(t)] \quad (1)$$

$$X_{i,j}(t+1) = X_{i,j}(t) + V_{i,j}(t+1), \ j = 1,2, \dots, d \quad (2)$$

W is the inertia coefficient; $C1$ and $C2$ are the cognitive coefficient and social learning coefficient, $R1$ and $R2$ denote random numbers between 0 and 1; $P_{i,j}$ is the local optimal particle position of the i^{th} particle in the j dimension space, $P_{g,j}$ is the global optimal particle position of the i^{th} particle in the j dimension space. PSO achieves its optimum solution by starting from a group of random solutions and then repeatedly searching [23, 24, 25, 26]. PSO has a good level of particle convergence because of the fast transmission of information among the particles. For this reason, swarm diversity decreases very quickly after the iterations and can lead to a suboptimal solution. This evolution process can trap in a local optimum or premature convergence.

Many variants of the PSO algorithm have been proposed to solve the diversity loss problem. The problem of decreasing diversity can be attributed to several factors. The population diversity of PSO is an important feature that demonstrates the exploration or exploitation ability of the algorithm. It is a technique used to determine the degree of convergence or divergence of PSO in the search process. As example, ARPSO is a method used to control the degree of diversity. It consists of an algorithm called ARPSO, which tests if the diversity is above the predefined threshold d_{low}, then particles attract each other, and if it is below d_{low}, then the particles repel each other until they meet the required high diversity d_{high}. LOD is also a method for local optima detectors; it consists of computing the number of iterations in which the neighbor does not improve, that is, if the fitness value (FV) of the best particle remains unchanged for a specific number of iterations, the particle optimization sub-process is trapped in a local optimum [27, 28, 29, 30, 31, 32]. To increase the diversity of swarms, several methods have been cited in the literature as particle re-initialization and particle mutation. Inspired from the idea of LOD, the authors apply this method to escape the worst particles from the local optimum.

C. Scheduling/ Rescheduling System

The goal of this system is to plan the production of a collection of jobs assigned to multiple machines given the production environment specifications. The scheduling problem has been demonstrated in the literature as a non-polynomial (NP-hard) [33]. In a multi-AGV system, n containers are available: $\{C_1, C_2, C_3, \dots C_n\}$ to be transferred by k AGVs$\{V_1, V_2, V_3, \dots V_k\}$, and the main objective is to determine the optimal schedule for n containers to be transported by the system. Each AGV can transfer only one container within a specific time interval. According to literature the selection of AGV can be based on one of the following methods: [34, 35]

- Longest travel distance
- Shortest travel distance
- Random
- Minimum AGV queue size

In its standard form, the scheduling problem can be described as a set of known tasks assigned to a set of available machines, considering the technological limitations of the system. This class of approach is called static scheduling. In this scheduling type, the tasks to be assigned and system parameters are known in advance and are invariant in time [36, 37, 38]. Multiple events, such as new task arrivals, machine breakdowns, task priority changes, and preventive machines, can affect the system in real situations. These changes in circumstances result from a dynamic environment that necessitates task reassignment. Rescheduling is defined as the process of updating existing production scheduling to react to any event. The literature explains three different strategies [39]. The Predictive-reactive strategy consists of providing an initial predictive schedule and changing it to reply to the disturbances that appear within the system. The proactive (or robust) strategy based on developing a schedule that absorbs any disturbances that may occur in the system. The dynamic strategy does not provide an initial schedule, but the assignment is performed dynamically. The authors choose the proactive rescheduling because the proposed approach was applied for static environment in first step, then it was extended for dynamic environment in second step.

III. Problem Formulation

Assume a set of containers $\mathrm{C} = \{\mathrm{C}_1, \mathrm{C}_2, \mathrm{C}_3, \ldots.. \mathrm{C}_\mathrm{m}\}$ stored in different locations $\mathrm{L} = \{\mathrm{L}_1, \mathrm{L}_2, \mathrm{L}_3, \ldots.. \mathrm{L}_\mathrm{n}\}$ at the port. These containers must be transferred to unloading locations to be transported by trucks and trains to clients, or inversely to a charging location to be loaded on a ship. A set of AGVs, $\mathrm{V} = \{\mathrm{V}_1, \mathrm{V}_2, \mathrm{V}_3, \ldots.. \mathrm{V}_\mathrm{k}\}$ should/will be available for transporting containers. The problem consists of assigning this set of containers to a set of AGVs and planning the path to each AGV. This problem can be decomposed into two sub-problems: A dispatching problem of AGVs to containers and a routing problem of AGVs. Many factors intervene in this optimization problem, such as distance traveled by AGVs, stability of the road network in the port, availability of AGVs, and utilization ratio of AGVs. For these reasons, the problem is considered as multi-objective problem. The authors choose to optimize the AGVs total travel distance and the balancing workload of AGVs. The AGVs dispatching and routing system can be assimilated to a scheduling system where a job is equivalent to the task of transferring the container from its origin location to its target location and the machine is equivalent to the AGV. In a static environment, the initial scheduling is sufficient for carrying a set of containers from their initial locations to their target locations. However, with the appearance of port automation, the number of pieces of equipment has become very important, and many events can appear and change the system situation. For example, new arrival of containers, changes in container priorities, breakdown of any equipment or AGV, AGV battery changes, and AGV preventive maintenance lead to system disruption. This change in the port situation requires system rescheduling. To develop an efficient scheduling system, it is necessary to study this problem in a dynamic environment. In first step, the authors study the problem of AGVs dispatching and routing in a static environment. In the second step they consider the cases of new containers arrival, the breakdown of AGVs, and the disruption of the road network in the port and proposed an extension of the first approach for resolving the problem in the dynamic environment.

A. Mathematic Model in a Static Environment

$C = \{C_1, C_2, C_3, \ldots.. C_m\}$: Set of containers.

$L = \{L_1, L_2, L_3, \ldots.. L_n\}$: Set of container locations (nodes).

$V = \{V_1, V_2, V_3, \ldots.. V_k\}$: Set of vehicles (AGV).

d_{ij} : Distance between nodes i and j

S : Speed of AGV

t_{ijk} : Travel time of vehicle V_k from node i to node j

$[tb_i, \ te_i]$: Time window of node i

tb_i : Beginning time of task in node i

te_i : Ending time of task in node i

td_i : Departure time from node i

ta_i : Arrival time at node i

tw_i : Waiting time at node i

S : Speed of AGV

q_{ik} : Load of AGV

Twt_k : Total work time of vehicle V_k

$$X_{ijk} : \text{Decision variable} \begin{cases} 1 \ \mathit{if\ the\ vehicle}\ V_k\ \mathit{is\ busy} \\ 0\ \mathit{otherwise} \end{cases} \quad (3)$$

The objective function is an aggregation of two sub-functions to be optimized: The function F1 for the total travel distance of all AGVs and the function F2 for the balance of AGVs workload.

$$F = \alpha * F_1 \ + \beta * F_2 \quad (4)$$

The objective function value depends on two coefficients α and β associated respectively to F1 and F2 which values are fixed by a domain specialist.

$$F \equiv \begin{cases} \mathrm{F}_1 \ = \ \sum_{\mathrm{i}\in\mathrm{N}} \sum_{\mathrm{j}\in\mathrm{N}} \sum_{\mathrm{k}\epsilon\mathrm{V}} \mathrm{X}_{\mathrm{ijk}} * \frac{\mathrm{d}_{\mathrm{ijk}}}{\mathrm{S}} \\ \mathrm{F}_2 = \sqrt{\left(\frac{1}{\mathrm{k}} - \frac{1}{\mathrm{k}^3}\right) * \left(\sum_{\mathrm{k}\epsilon\mathrm{V}} \mathrm{Twt_k}^2 - 2 * \prod_{\mathrm{k}\epsilon\mathrm{V}} \mathrm{Twt_k}\right)} \end{cases} \quad (5)$$

$$\sum_{i\epsilon N} \sum_{k\epsilon V} X_{ijk} \ = 1, \forall \ j \in N \quad (6)$$

$$\sum_{i\epsilon N} X_{i0k} \ = 1, \forall k \in V \quad (7)$$

$$\sum_{j\epsilon N} X_{0ik} = 1, \forall k \in V \quad (8)$$

$$\sum_{i\epsilon N} X_{ijk} - \sum_{i\epsilon N} X_{jik} = 0, \forall k \in V, \forall j \in N \quad (9)$$

$$Q_k = \sum_{i\epsilon N} \sum_{j\epsilon N} q_{ijk} = 1, \forall k \in V \quad (10)$$

$$X_{ijk} = 1 \rightarrow tb_i \leq ta_i < te_i, \forall i \in N, \forall j \in N, \forall k \in V \quad (11)$$

$$X_{ijk} = 1 \rightarrow tb_i \leq td_i < te_i, \forall i \in N, \forall j \in N, \forall k \in V \quad (12)$$

$$X_{ijk} = 1 \rightarrow ta_i - t_{ijk} + Tw_i \leq_i tb_i, \forall i \in N, \forall j \in N, \forall k \in V \quad (13)$$

$$X_{ijk} = 1 \rightarrow td_i - t_{ijk} + Tw_i \leq_i te_i, \forall i \in N, \forall j \in N, \forall k \in V \quad (14)$$

(6) The transport cost from node i to node j is equal to 1.

(7) and (8): the possibility of moving from node zero to any other node.

(9): bi-directionality of each edge.

(10): AGV load equals 1

(11): The AGV must arrive at node i within the arrival time window.

(12): The container must be moved within the departure time window of the node.

(13) and (14): The AGV must arrive before the beginning of the node time window, and must move before the end of the node time window.

B. Dynamic Environment Parameters

In containers terminal, the real situation is dynamic and uncertain. Any equipment such as quay crane, truck, AGV, road can breakdown at any moment of time. A lateness of the loading/unloading operation for the corresponding ship can appear. This tardiness will propagate for all the ships coming after. Three cases was investigated in this study:

a) New Arrival Containers: This case can change the number of containers. Assume C' the set of containers of the new arriving ship. For the current process, the total number of containers to be loaded/unloaded will be $C_T = C \cup C'$. Each AGV will have an extra number of containers to transfer.

$C = \{C_1, C_2, C_3, \ldots\ldots C_m\}$: set of containers for current ship

$C' = \{C'_1, C'_2, C'_3, \ldots\ldots C'_n\}$: set of containers for new ship

$C_T = \{C_1, C_2, C_3, \ldots\ldots C_m, C'_1, C'_2, C'_3, \ldots\ldots C'_n\}$: Total set of containers to be loaded/unloaded

b) Road Network Disturbance: This case appear when there is a breakdown in some nodes of road network. Assume L1 and L2 are unavailable, all the paths containing these two nodes will be modified. Assume $L = \{L_1, L_2, L_3, \ldots\ldots L_n\}$, the initial set of nodes, the new set of nodes will be $L' = L \backslash \{L_1, L_2\}$. As consequence the AGV will travel a path other than the shortest path proposed initially.

c) AGVs Breakdown: If an AGV is unavailable, the set of containers corresponding to this AGV will be assigned to other AGVs. Assume $V = \{V_1, V_2, V_3, \ldots\ldots V_k\}$, if V_1 and V_3 are unavailable, the new set of available AGVs will be $V' = V \backslash \{V_1, V_3\}$. This event will have an effect on the workload balance of AGVs.

IV. Proposed Approaches

To increase the diversity of swarms, several methods have been cited in the literature as particle re-initialization and particle mutation. The main of particle re-initialization method is to increase the possibility of "jumping out" of local optima and to maintain the ability of the algorithm to find the "good enough" solution. After several iterations, some particles were selected to reinitialize their position and velocity. The number of chosen particles can be either constant or fuzzy. Three methods to select particles: 1) The random selection consists of selecting randomly a set particles to reinitialize its position and velocity. This method can obtain great exploration ability owing to the possibility that all particles have the chance to be reinitialized. 2) The elitist selection based on choosing a set of the best particles, having the best fitness value, to reinitialize its position and velocity. When the population diversity decreases, most particles have the best fitness values. When these elitists are reinitialized, the exploration ability of the algorithm increases but the good particles can be lost. 3) The worst particle selection consists of choosing a set of the worst particles to reinitialize its position and velocity. This idea can increase the ability of the algorithm to explore space and find the "good enough" solution by ameliorating bad particles. The particle mutation method is based on applying the mutation feature of GA to change one or more features of a particle to achieve better particles quality. This is a common method for increasing the population diversity. It can improve exploration abilities, which can be applied to different elements of a particle swarm.

The authors propose a new hybrid PSO approach called HPSO based on a heuristic and PSO algorithm. This approach did not show a clear convergence. It presents a very quick convergence which risks premature convergence. To improve the approach results, the authors propose a second approach guided particle swarm called GPSO. It consists of guiding the HPSO approach in the routing problem by choosing the shortest path for each AGV by applying the Dijkstra algorithm. This approach presents acceptable convergence. In comparing its results with previous results, it appears acceptable, but it's necessary to verify the problem of premature convergence. A third approach is proposed, named GHPSO, which combines the ameliorations of the two previous approaches (Fig. 1).

A. Hybrid Particle Swarm Approach (HPSO)

This approach is a hybridization between a heuristic, the Dijkstra algorithm, and the particle swarm algorithm; it is called the HPSO approach**.** It uses a heuristic based on assigning each container to the nearest AGV.

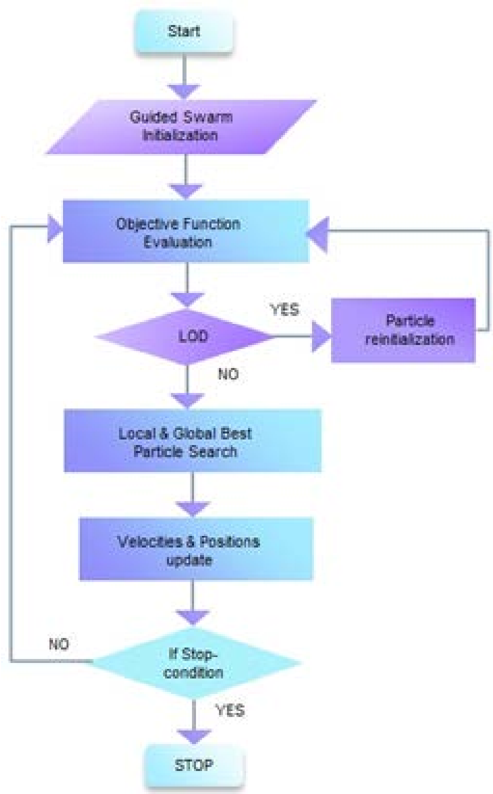

Fig. 1. GHPSO Flowchart.

The AGV travels the shortest distance to arrive at a container location. The shortest path problem is solved using the Dijkstra algorithm, and the optimal solution is determined by applying the PSO algorithm through a fixed number of iterations. The HPSO algorithm is as follows (Fig. 2).

Algorithm1: HPSO
Results: GBest
Iterations = 1;
Guided_Swarm_Initialization( );
While (Iterations <= Nb_iterations)
{
Objective_Function_Evaluation( );
Best_Positions_Search( );
Particles_Updates( );
Iteration = Iteration + 1;
}
End Algo
Procedure1: Guided_Swarm_Initialization()
Sort_List_Containers( );
While (List_Containers # ϕ)
{
Assign_container_AGV( );
Choose_shortest_path( );
}
End proc

Fig. 2. HPSO Algorithm.

B. *Guided Hybrid particle Swarm Approach (GHPSO)*

The HPSO approach and the GPSO approach show a fast convergence which risks the premature convergence. To surpass this deficiency, the authors propose to study the diversity of PSO population. The population diversity was computed to prevent premature convergence. They choose to control the activity of particles to detect which particles were responsible for the diversity of population loss. LOD (local optimum detector) for each particle to determine whether the particle is inactive for an important number of iterations. The authors selected a threshold value for the number of iterations.

After detecting these particles, they re-initialize the positions and velocities for all particles to jump out of the local optimum.

C. *Robustness of the GHPSO in a Dynamic Environment*

In a dynamic container terminal environment, any disturbance can cause an increase in the number of containers in depots or Quays, because the waiting time for loading or unloading increases. The number of AGVs in the port is fixed but may decrease due to any breakdown (Fig. 3).

Algorithm2: GHPSO
Results: GBest
Iterations = 1;
Guided_Swarm_Initialization( );
While (Iterations <= Nb_iterations)
{

While (List_Particles # ϕ)
if (LOD(Particle) == true)
{

Particle_reinitialization( );
Go to **EV**;
}
else
{
Best_Positions_Search( );
Particles_Updates( );
}
Iteration = Iteration + 1;
}
Procedure2: LOD(Particle, limit_repetition)
if (nb_repetition_Particle == limit_repetition)
{
nb_repetition_Particle = 0;
return true;
}
else
{
nb_repetition_Particle = nb_repetition_Particle + 1;
return false;
}
End proc

Fig. 3. GHPSO Algorithm.

End proc

The authors propose an extension of the GHPSO approach for a dynamic environment, dynamic guided hybrid particle swarm called DGHPSO approach, which studies three disturbance cases:

a) Arrival of New Containers: To solve the problem of new container arrivals, DGHPSO proposes to add the new containers to the AGVs queues during the Algorithm execution. The number of containers associated to each AGV will increase. An AGV_i can begin the dispatching process with n containers and finishes it with n+m containers. The solution will be optimized after completing the iterations.

b) AGV Breakdown: The DGHPSO approach proposes a new distribution of containers associated with a broken AGV for other AGVs. The number of containers for the available AGVs increases. New dispatching was proposed and optimized after completing the iterations.

c) Network Road Disruption: The unavailability of any node in the road network affects the set of paths proposed for the AGVs. This disturbance can cause inaccessibility of any path. AGVs must stop the transfer of the associated container. The DGHPSO approach proposes a new path to travel, then the solution will be optimized.

V. Experimental Studies

The application of these approaches was performed with a computer having 8 GOs of RAM and a processor speed of 2.4 GHz. The proposed approaches were implemented with a swarm population of 50 particles, the number of AGVs is 4 and the number of containers is 20. The PSO parameter values chosen after several tests were C1=2, C2=2, Wmin=0.4, and Wmax=0.9. R1 and R2 were randomly chosen such that $R_1 +$ R2 = 1. The threshold value chosen after several tests was 5. After several numeric tests, the genetic algorithm parameters chosen are as follows: 70% of the population was selected for crossing over and 10% for mutation.

Numerical tests were applied to two previous versions of the GA approach [40] and four new versions of the PSO approach.

Fig. 4 shows a comparative graph between the two previous genetic algorithm approaches: genetic algorithm (GA) and hybrid genetic algorithm (HGA: GA + Dijkstra). The initial solutions of the two approaches are very different. This demonstrates the importance of hybridization with the Dijkstra algorithm in the second approach. The convergence of the GA with an optimum solution value is 2.51, but it is not significant because of the random paths chosen for the AGVs. The HGA graph shows good convergence with an optimum solution value of 1.5 because the paths are optimized using the Dijkstra algorithm.

Fig. 5 presents a comparative graph of the PSO approaches. The basic PSO algorithm (PSO curve in blue) shows quick convergence from the first iterations with an optimum solution value of 2.6, and it becomes almost stable at 2.5.

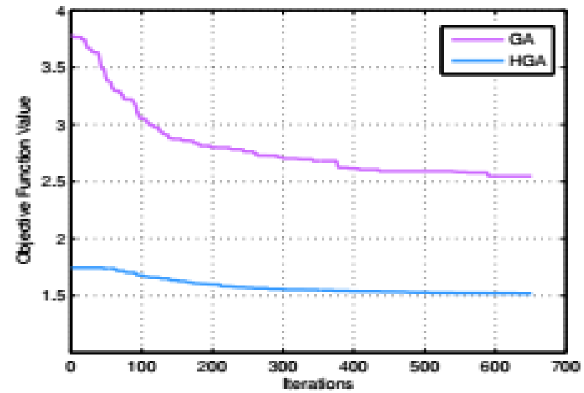

Fig. 4. GA Approaches Comparison.

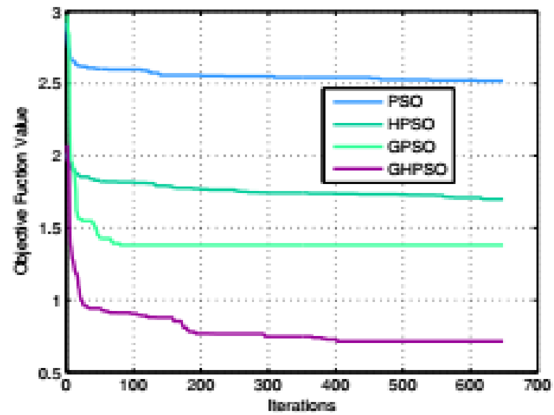

Fig. 5. PSO Approaches Comparison.

The hybridization with Dijkstra's algorithm and the insertion of a heuristic for choosing the nearest AGV with the standard PSO (HPSO curve in green) considerably improves the optimum solution value from 2.5 to 1.7, but the convergence is again not very remarkable. It is clear that the population diversity quickly decreases. The guided particle swarm approach GPSO (PSO+ re-initialization: curve in light green) shows a slight improvement in the solution value. Its value decreases from 1.7 to 1.4, and the convergence appeared significant. The last curve represents the GHPSO model. Its solution value is 0.7, which is the optimum among all proposed approaches. This convergence becomes significant in comparison with previous approaches. It is clear that the problem of a faster decrease in population diversity is solved by the insertion of LOD and the re-initialization of particles.

To determine the best approach, the authors performed a final comparison between the best GA and PSO approaches, as shown in Fig. 6. The graphs show good convergence of the PSO approach in comparison with the previous GA approach.

A comparison of the running times computed for each approach is shown in Fig. 7. The GHPSO presented an acceptable running time of 19.510-3 s in comparison with other proposed approaches.

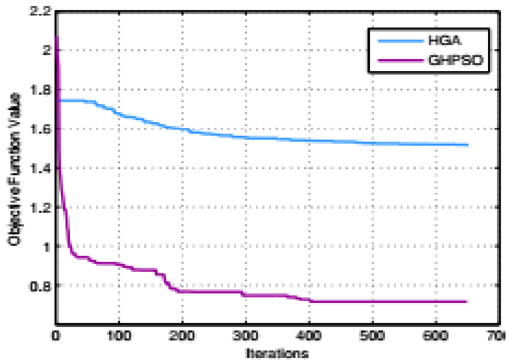

Fig. 6. GA and PSO Aproaches Comparison.

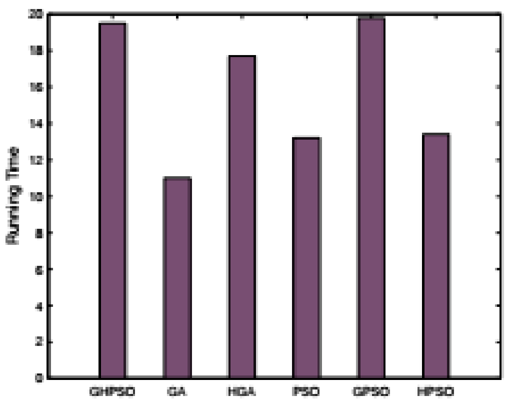

Fig. 7. Running Time Comparison.

The numerical results show the good performance of the GHPSO as the best PSO approach in a static environment. This approach also surpasses the performance of GA. This deduction encourages the authors to apply this approach in a dynamic environment, where the number of containers, AGVs, and nodes in the road network are not fixed.

Fig. 8 shows the approach convergence after the insertion of new containers. The authors propose the insertion of 20 containers at 50 iterations. It is clear that the approach begins by improving the initial solution to determine the optimum solution. The initial solution objective function value is 3, and at iteration 50, it becomes 1. When new containers were inserted, the solution value increased to 1.65. Subsequently, it was again in decreasing order to find the best solution. It reached notable convergence at almost 120 iterations with an optimum solution value of 0.6. This result demonstrates the robustness of the GHPSO approach in determining the optimum solution in the case of new container arrivals.

Fig. 9 presents the numerical results of an AGV breakdown at iteration 100, when the guided hybrid particle swarm approach (GHPSO) begins to converge. The solution value increases again because of the distance traveled by each AGV, but after 140 iterations, it converges again with a solution value of 1.25, which was greater than the initial optimum solution value due to increasing of workload of each AGV. These results demonstrate the GHPSO approach efficiency in reaching the optimum solution.

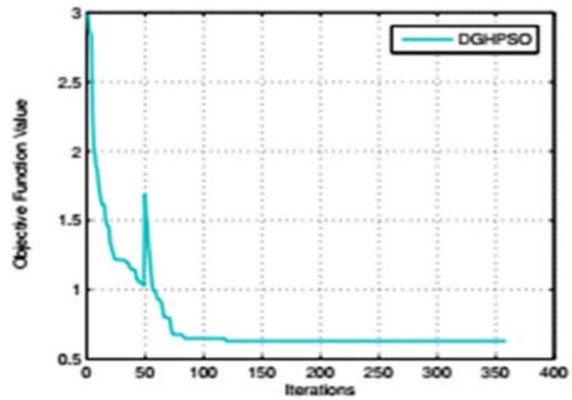

Fig. 8. New Containers Arrival.

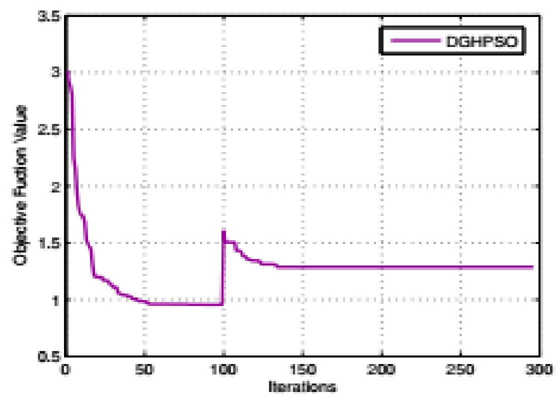

Fig. 9. AGV Breakdown

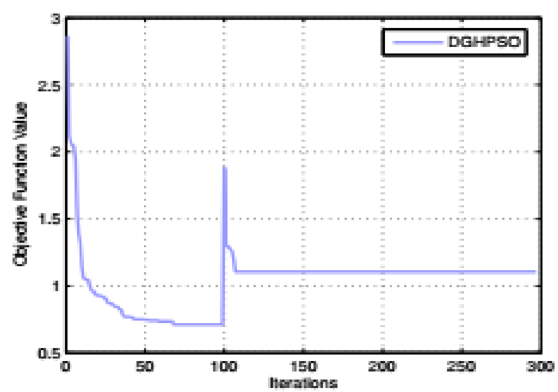

Fig. 10. Unavailable Path.

Fig. 10 shows the numerical results when some paths become unavailable. The solution value increased again and then decreased to converge to a solution value of 1.2, which was greater than the optimum solution before path breakdown due to unavailability of shortest path.

VI. Conclusion and Perspectives

In this study, the problem of dispatching containers to AGVs in a container terminal in static and dynamic environments was investigated using an improved PSO algorithm. A new guided PSO approach named GHPSO was proposed for optimizing the total travel distance and balancing the workload between AGVs. The first idea was to combine heuristic and Dijkstra with the PSO algorithm to obtain the optimum solution. The numerical results show an acceptable solution, but the convergence is not significant because an important number of particles cannot ameliorate their best local solution. The idea was to apply the insertion of the LOD parameter and reinitialize the particles to achieve good results. The convergence became significant for the GPSO, and the best optimum solution value was obtained using the GHPSO. This work showed a very good GHPSO performance compared to other approaches, although the running time was acceptable. The proposed approach was tested in a dynamic environment (DGHPSO), where the number of containers, number of AGVs, and network road nodes were not fixed. To demonstrate the robustness of this approach, the authors proposed an extension to study the arrival of new containers during approach execution. The numerical results show good convergence for the approach. In addition, for the two AGV breakdown and node breakdown cases, the proposed approach shows good convergence. This approach shows good robustness in static and dynamic environments for finding the optimum solution within a reasonable running time. In future work, the authors will study the efficiency of this approach for the multi-objective problem with the AGV energy constraint and its effect on task allocation in a static and uncertain environment.

ACKNOWLEDGMENT

The authors gratefully acknowledge the approval and the support of this research study by the grant no SAO-2018-3-9-F-7766 from the deanship of scientific research at Northern Border University, Arar, Kingdom of Saudi Arabia.

REFERENCES

[1] Zakaria Z, Petrovic S. (2012). Genetic algorithms for match-up rescheduling of the flexible manufacturing systems. Computers & Industrial Engineering, 62(2), 670-686.

[2] Oyekanlu EA, Smith AC, Thomas WP, Mulroy G, Hitesh D, Ramsey M, Sun D. (2020). A review of recent advances in automated guided vehicle technologies: Integration challenges and research areas for 5G-based smart manufacturing applications. IEEE access, 8, 202312-202353.

[3] De Ryck M, Versteyhe M, Debrouwere F. (2020). Automated guided vehicle systems, state-ofthe-art control algorithms and techniques. Journal of Manufacturing Systems, 54, 152-173.

[4] Bae J, Chung W. (2018). A heuristic for path planning of multiple heterogeneous automated guided vehicles. International Journal of Precision Engineering and Manufacturing, 19(12), 1765-1771.

[5] Luo J, Wu Y. (2020). Scheduling of container-handling equipment during the loading process at an automated container terminal. Computers & Industrial Engineering, 149, 106848.

[6] Mousavi M, Yap HJ, Musa S N, Dawal SZM. (2017). A fuzzy hybrid GA-PSO algorithm for multi-objective AGV scheduling in FMS. a a, 1(1), 1.

[7] Fazlollahtabar H, Saidi-Mehrabad M. (2015). Methodologies to optimize automated guided vehicle scheduling and routing problems: a review study. Journal of Intelligent & Robotic Systems, 77(3), 525-545.

[8] Kizilay D, Eliiyi DT. (2021). A comprehensive review of quay crane scheduling, yard operations and integrations thereof in container terminals. Flexible Services and Manufacturing Journal, 33(1), 1-42.

[9] Vancea AP, Orha I. (2019). A survey in the design and control of automated guided vehicle systems. Carpathian Journal of Electronic and Computer Engineering, 12(2), 41-49.

[10] Hyla P, Szpytko J. (2017). Automated guided vehicles: the survey. Journal of KONES, 24.

[11] Yang Y, Zhong M, Dessouky Y, Postolache O. (2018). An integrated scheduling method for AGV routing in automated container terminals. Computers Industrial Engineering, 126, 482-493.

[12] Zhang Q, Hu W, Duan J, Qin J. (2021). Cooperative Scheduling of AGV and ASC in Automation Container Terminal Relay Operation Mode. Mathematical Problems in Engineering, 2021.

[13] Chawla V, Chanda A, Angra S. (2018). Scheduling of multi load AGVs in FMS by modifed memetic particle swarm optimization algorithm. Journal of Project Management, 3(1), 39-54.

[14] Gola A, K losowski G. (2017, June). Application of fuzzy logic and genetic algorithms in automated works transport organization. In International symposium on distributed computing and artificial intelligence (pp. 29-36). Springer, Cham.

[15] Mishra N, Roy D. and, van Ommeren JK. (2017). A stochastic model for inter terminal container transportation. Transportation science, 51(1), pp.67-87.

[16] Li MP, Kuhl ME. (2017, December). Design and simulation analysis of PDER: A multiple load automated guided vehicle dispatching algorithm. In 2017 winter simulation conference (wsc) (pp. 3311-3322). IEEE. Scheduling of container-handling equipment during the loading process at an automated container terminal.

[17] Han Z, Wang D, Liu F, Zhao Z. (2017). Multi-AGV path planning with double-path constraints by using an improved genetic algorithm. PloS one, 12(7), e0181747.

[18] Hu X, Chen L, Tang B, Cao D, He H. (2018). Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles. Mechanical systems and signal processing, 100, 482-500.

[19] Zhou P, Lin L, Kim KH. (2021). Anisotropic Q-learning and waiting estimation based real time routing for automated guided vehicles at container terminals. Journal of Heuristics, pp.1-22.

[20] Yue L, Fan H, Ma M. (2021). Optimizing configuration and scheduling of double 40 ft dual trolley quay cranes and AGVs for improving container terminal services. Journal of Cleaner Production, 292, p.126019. 17.

[21] Gu W, Li Y, Zheng K, Yuan M. (2020). A bio-inspired scheduling approach for machines and automated guided vehicles in flexible manufacturing system using hormone secretion principle. Advances in mechanical engineering, 12(2), p.1687814020907787.

[22] Yan X, Zhang C, Luo W, Li W, Chen W, Liu H. (2012). Solve traveling salesman problem using particle swarm optimization algorithm. International Journal of Computer Science Issues (IJCSI), 9(6), 264.

[23] Cheng S, Sh Y. (2011, April). Diversity control in particle swarm optimization. In 2011 IEEE Symposium on Swarm Intelligence (pp. 1-9). IEEE.

[24] Ding J, Liu J, Chowdhury KR, Zhang W, Hu Q, Lei J. (2014). A particle swarm optimization using local stochastic search and enhancing diversity for continuous optimization. Neuro computing, 137, 261-267.

[25] Bonyadi MR, Michalewicz Z. (2017). Particle swarm optimization for single objective continuous space problems: a review. Evolutionary computation, 25(1), 1-54.

[26] Freitas D, Lopes LG, Morgado-Dias F. (2020). Particle swarm optimization: a historical review up to the current developments. Entropy, 22(3), 362.

[27] Li N, Li YX. (2012). A Diversity Guided Particles Swarm Optimization. In Advanced Materials Research (Vol. 532, pp. 1429-1433). Trans Tech Publications Ltd.

[28] Han F, Liu Q. (2014). A diversity-guided hybrid particle swarm optimization based on gradient search. Neuro-computing, 137, 234-240.

[29] Wang D, Tan D, Liu L. (2018). Particle swarm optimization algorithm: an overview. Soft Computing, 22(2), pp.387-408.

[30] Abdel-Kader RF. 2018. An improved PSO algorithm with genetic and neighborhood-based diversity operators for the job shop scheduling problem. Applied Artificial Intelligence, 32(5), pp.433-462.

[31] Ayari A, Bouamama S. (2019). ACD3GPSO: automatic clustering-based algorithm for multi-robot task allocation using dynamic distributed double-guided particle swarm optimization. Assembly Automation.

[32] Song Z, Liu B, Cheng H. (2019). Adaptive particle swarm optimization with population diversity control and its application in tandem blade optimization. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 233(6), 1859-1875

[33] Ahmadian, M.M., Khatami, M., Salehipour, A. and Cheng, T.C.E., 2021. Four decades of research on the open-shop scheduling problem to minimize the makespan. *European Journal of Operational Research*, *295*(2), pp.399-426.

[34] Chin IL, Ioannou PA. (2002). A Comparison of Different AGV Dispatching Rules in an Automated Container Terminal. The IEEE 5th International Conference on Intelligent Transportation Systems, IEEE, Si.

[35] Heger, Jens, and Thomas Voß. "Dynamic priority based dispatching of AGVs in flexible job shops." *Procedia CIRP* 79 (2019): 445-449.ngapore, pp. 880-885.

[36] Valledor P, Gomez A, Priore P, Puente J. (2018). Particle swarm optimization algorithm: an overview. Soft Computing, 22(2), 387-408.

[37] Cai, B., Huang, S., Liu, D. and Dissanayake, G., 2014. Rescheduling policies for large-scale task allocation of autonomous straddle carriers under uncertainty at automated container.

[38] Xin, J., Negenborn, R.R. and Lodewijks, G., 2014. Rescheduling of interacting machines in automated container terminals. *IFAC Proceedings Volumes*, *47*(3), pp.1698-1704.

[39] Kuster J, Jannach D, Friedrich G. (2010). Applying local rescheduling in response to schedule disruptions. Annals of Operations Research, 180(1), 265-282.

[40] Zaghdoud, R., Mesghouni, K., Dutilleul, S. C., Zidi, K., Ghedira, K. A Hybrid Method for Assigning Containers to AGVs in the Dynamic Environment of Container Terminals. Studies in informatics and control, 2015, Vol. 24, No. 1, pp. 43-50. ISSN 1220-1766.