Multi AGV path planning

***Abstract*—scheduling and planning are amongst the most important factors in the automation of workshops.This paper attempts to study the multi-robot path and schedule planning problems using different techniques to asses which is optimal to be used in a workshop based on multi-line following AGV's**

***Index Terms*—Optimization, meta-heuristic,Multi-AGV,path-planning,simulated-annealing,Genetic algorithm,particle-swarm-optimization,African-buffalo-optimization.**

I. Introduction

A. Multi-AGV

Recent trends in manufacturing, storage, and distribution plants are going towards full automation using robots similar to AGVs connected to an FMS that schedules and directs the movements of the robots to perform the wanted tasks. An important factor of this system is to minimize the time and cost of operation by optimizing the paths which the AGV takes. This is important to assure maximum efficiency of operation, especially when dealing with multiple AGVs at the same time.

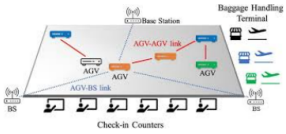

Fig. 1: AGV Path planning

B. Optimization techniques

Optimization techniques have been an emerging research field due to their importance in many fields today.Optimization is widely defined as finding an alternative with the most cost effective or highest achievable performance under the given constraints, by maximizing desired factors and minimizing undesired ones. The problem of optimization can be formulated by choosing wisely the decision variables, the objective function, and the constraint function in order to maximize desired factors and minimize desired ones. The solution to many problems around us has been addressed using many optimization techniques.

II. Literature review

As stated earlier, optimization techniques have been an emerging field due to its importance in many fields today. Many researches have been conducted on our specific problem which is Multi AGV path planning. Mohamad *et al...* use genetic optimization algorithm (GA) to find an optimal solution for an FMS layout representing a model of a multi-AGV system for a material transport assignment problem.The results of the paper show a competitive time in regards to most outcomes especially reducing total vehicle travel considering material load time [1]

Wu *et al...* proposed using a hybrid solution between Water Wave Optimization (WWO) and Tabu Search (TS) algorithms to solve the multi-AGV problem. The optimization algorithm is composed of two primary levels. The main inner level of the algorithm is to use the TS method to find a population of optimized solutions which would be the input for the second level of the algorithm. The second Level would use the WWO method to find the main optimized solutions as an output. The findings of the paper show a variable substantial improvements compared to other meta-heuristic techniques. [2].
Xiaohua Cao *et al...* tackles the path conflicts that occur in multi-AGV systems often. The paper proposes an optimization method of AGV traffic sequence arranging.Improved PSO algorithm proved effective in reducing congestion. [3].
QIUYUN *et al...* discussed how Particle Swarm Optimization (PSO) algorithm could be useful in such a problem, especially when taking into consideration the collision between AGVs. Also, QIUYUN proposed an improved PSO Algorithm that considers the collisions between Multi-AGVs. [4]

III. Methodology

A. problem Formulation

The problem of AGV path optimization is usually described as: In a warehouse, workshop, or factory. There are AGVs (n) and shelves/production lines (m). the number of shelves shall be greater than the number of AGVs transporting. m ¿ n. The AGVs shall receive orders from the FMS and select the best path in terms of the shortest distance to be covered by the AGV in order to transport material from/to shelves/production lines. This makes the objective function for us is to minimize the distance covered by each AGV and the time taken by each AGV. This objective function can be represented by the following equations:

$$F = min(\Sigma_{i=1}^{m} distance) \tag{1}$$

$$F = min(time = timebyeachrobot + collisions) \tag{2}$$

$$cost = distance + time \tag{3}$$

The next figure shows a demonstration of the multi-AGV path planning problem and the layout of the warehouse/factory. The layout of the map in our optimization algorithm will be given in the shape of line following for the AGVs.

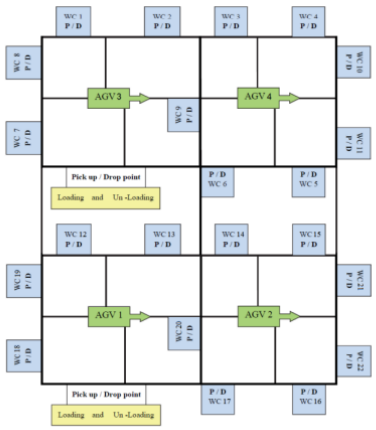

Fig. 2: multi AGV layout

In order to formulate our problem correctly. The Optimization problem must have several assumptions made first in order to start executing algorithms These assumptions are:

- AGV speed is consistent during operation
- All equipment and machines are available
- The distance between all shelves is equal
- AGV does not need to return to the starting point after performing tasks, but directly executes the next task in the queue, and does not return to the warehouse until all tasks are complete
- Maps of factories which contain all distances and paths are pre-given in our problem which assumed to be handed by the factory itself
- no recharging at stations
- All AGVs are reliable and unbreakable
- All AGVs are bidirectional
- Time of Loading/unloading will be considered constant

Our Optimization algorithm will consist of key factors which will be considered as the main elements for our study and our analysis.

- Cost: is what it takes to perform a certain task and is not necessarily dependent on money. It could be time, distance, fuel consumption or battery usage, etc.
- Fitness: is how well a certain task could be done by a certain candidate, this could be how well a certain robot or machine can perform a task
- Priority: is the necessity or urgency of doing a particular task

The orders will be received by the FMS and contain which workstation to load/unload from. The orders will be given via array and each cell has the workstation number which is known before for the AGV to go to. The decision variables for this optimization algorithm are:

- The number of AGVs that will go and hand packages
- Order of workstations
- which AGV will deliver to which station

Fig. 3: orders example

The constraints are:

- The number of AGVs available in the Factory
- map size
- Position of workstations
- maximum amout of packages that can be carried by each robot

In our specific problem. Simulated annealing algorithm, Genetic Algorithm, Discrete particle swarm oprimization and African Buffalo optimization were all implemented to test and see the results and which optimization technique yielded the best results in terms of cost function which means min(distance,time). Also, 3 other performnace metrics were applied to test and see the effect of each optimization technique on our problem.

B. Collisions

The collision between AGVS with other AGVS will be computed and time will be increased by 1 sec for every collision betweeen AGVs. time will be added to the cost function.

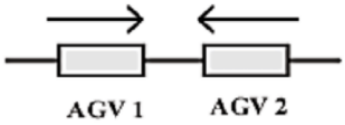

Fig. 4: collsion example

The collision optimization in our problem was done using the penalty technique pseudo code:

- check current position
- if next pos AGV1 = next pos AGV2
- penalty=penalty+1
- time=time+penalty

C. Simulated annealing algorithm

Simulated annealing is a method for solving unconstrained and bound-constrained optimization problems. The method models the physical process of heating a material and then slowly lowering the temperature to decrease defects, thus minimizing the system energy.The SA algorithm was implemented to be scalable . e.g : number of workstations can be changed, number of AGVs can be changed, map size can be changed. in our implemented design the SA algorithm was as follows:

- map size was set
- initialize Start and Goal points
- Initialize the optimization problem parameters
- Initialize a random solution
- set old and best solutions for first iteration
- generate new Solution and path that comes with solution
- if not feasible generate new solution

- check if better solution if not check if acceptable solution
- if current is better than global best Update Best
- Update time
- visualization

The random assignment were as follows:
having an array of variable size we need to assign randomly the order of which workstation to go to first. Also, with random assignment of how much each robot will take orders, by taking into considerations of course the weight limit of each robot. then applying the swap function to make the new solution depend on the old solution and not generate a new solution every time as this is a main aspect of the simulated annealing algorithm.

D. Genetic Algorithm

Natural selection, the mechanism that propels biological evolution, is the foundation of the genetic algorithm, a technique for resolving both limited and unconstrained optimization issues.An individual population of solutions is repeatedly altered by the genetic algorithm.

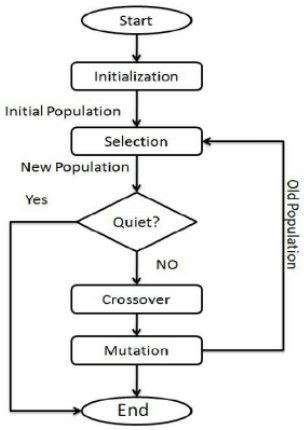

Fig. 5: GA flowchart

1) Crossover: The crossover was done in the GA by the following algorithm of Davis order crossover: first we begin by selecting two crossing points and then we calculate the positions from the second point to the first. Then we decide on the Arrangement for children. Moving on the the finalization of values between the two crossing points then Populating the first child from the second crossover point without it belonging to the other. parent,Order from the second point to the same

2) Mutation: The mutation uses the swap method which was in the SA algorithm.

3) Elite: The Elite were chosen with the best cost function with a specific percentage

4) Parent selection: All the population were sorted with respect to the cost function. Then the every parent is selected with the parent beneath it. and if they are odd number the last parent will cross over with the previous parent.

E. Discrete Particle swarm optimization

Kennedy and Eberhart (1995) developed the Particle Swarm Optimization (PSO) algorithm as a meta-heuristic algorithm based on the social behavior exhibited by bird flocks when striving to reach a destination. The analogy of the PSO algorithm is as follows

Optimization Algorithm	PSO Algorithm
Solution	Particle Position
Objective Function	Distance between Particle and Food
Old Solution	Old Particle Position
New Solution	New Particle Position
Best Solution	Leader
Generation of Solutions	Flying with specific velocity

The PSO algorithm can only deal with Arithmetic problems thus the discrete PSO technique was implemented to match our problem specifications as our problem is considered a permutation problem.

1) Discrete PSO motion update: In this application of PSO we are using Discrete PSO Because our problem is a permutation problem.The motion update technique uses the star topology with constant inertia weight then after updating the position the solution is discretized.

2) Discretization Pseudo-Code:

- generate an array that contains all required nodes ordered from smallest to largest
- get the order of the current position
- create a new solution based on the order of the current position (decending order)

3) Discrete PSO motion update:

$$v_{i+1} = wv_i + c1 * R1(Pbest_i - x_i) + c2 * R2(Nbest_i - x_i) \quad (4)$$

$$x_{i+1} = x_i + v_{i+1} \quad (5)$$

F. African buffalo optimization

The African Buffalo Optimization is a newly designed metahauristic optimization algorithm inspired by the migration of African buffalos from place to place across the vast African forests, deserts and savannah in search of food. The ABO first initializes all buffalos to cerrain random location in the search space. Then fitness is evaluated in terms of exploration and exploitation of each buffalo to determine the herds best buffalo (bg) and (bp) the individual buffalo personal best locations.

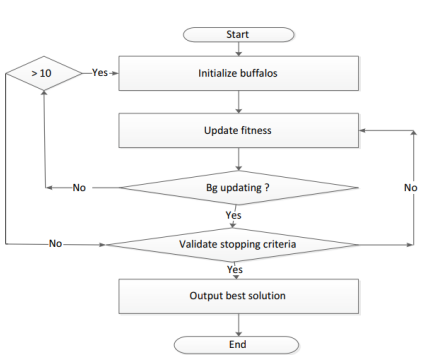

Fig. 6: ABO flowchart

Te controlling equations that propels the entire buffalo herd to relocate to other locations, probably more rewarding than the present location are:

$$m_{k+1} = mk + lp1(bg - w_k) + lp2(bp.k - w_k) \quad (6)$$

$$w_{k+1} = (w_k + m_k)/lamda \quad (7)$$

The w_k represents the waaaa calls (move on/explore) with particular tefrence to buffalo k. This call is a signal to the buffalos to move on to a safer or more rewarding locatio. mk denotes the maaa call (stay to exploit) that requests the buffalos. w_{k+1} denotes the request for further exploration. similarly,m_{k+1} denotes for futher exploitation. lp1 and lp2 are learning parameters, lamda is a random number thta takes a value of between 0 and 1.

1) ABO pseudo code:

- Initialize buffalos randomly at search space
- update buffalos exploitation using equation 6
- update location of buffalos using equation 7
- is bgmax updating ? Yes go to 6. If no, go to 1
- Validate stopping criteria, Reached, go to 6. Otherwise return to step 2
- Output best solution

IV. Results

A. SA results

1st case study we will be analyzing is with the following parameters

parameter	value
Map size	100
InitalPosition	6
Workstations	15
NumberOfRobots	6
MaximumWorkstationperRobot	10
Tinitial	500
Beta	1
Max.iterations	(Tinit-Tfinal)/Beta

2nd case study we will be analyzing is with the following parameters

parameter	value
Map size	100
InitalPosition	10
Workstations	10
NumberOfRobots	10
MaximumWorkstationperRobot	10
Tinitial	750
Beta	1
Max.iterations	(Tinit-Tfinal)/Beta

The results are shown in the next figures:

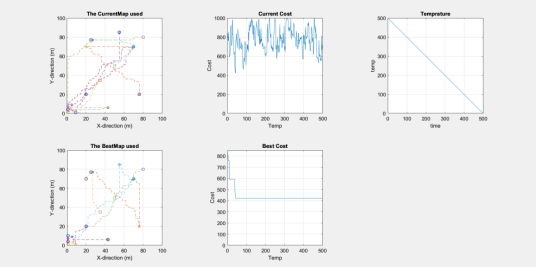

Fig. 7: Case study 1 results

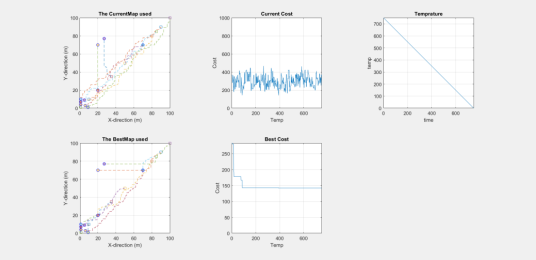

Fig. 8: Case study 2 results

In case study 1 we have 6 AGVs while in case study 2 we have 10 AGVs. We can observe here that the cost in case study 2 has decreased. Also, in case study 2 we have less workstations with ratio to AGVs. nearly 1 workstation/AGV while in case study 1 we have 2.5 workstations/AGV.

B. GA results

case study 1

parameter	value
InitalPosition	6
workstations	15
Numgenerations	60
Popsize	100
Eliteratio	0.2
CrossOverratio	0.6
Mutationratio	0.2

case study 2

parameter	value
InitalPosition	10
workstations	10
NumGenerations	60
Popsize	100
Eliteratio	0.2
CrossOverratio	0.6
Mutationratio	0.2

The results are shown in the following figures:

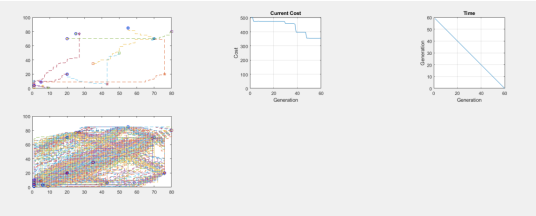

Fig. 9: Case study 1 results

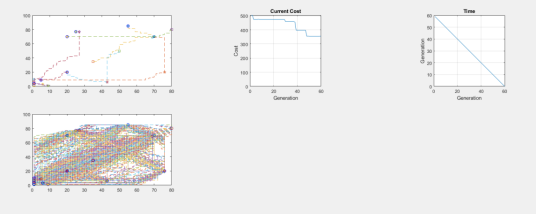

Fig. 10: Case study 2 results

C. Discrete PSO results

Case Study 1

parameter	value
InitalPosition	6
workstations	15
swarm size	80
Numiterations	100
inertia weight factor	0.9
acceleration coefficient due to personal best	0.9
acceleration coefficient due to global best	0.1

Case Study 2

parameter	value
InitalPosition	10
workstations	10
swarm size	80
Numiterations	100
inertia weight factor	0.9
acceleration coefficient due to personal best	0.9
acceleration coefficient due to global best	0.1

Case study 1 results:

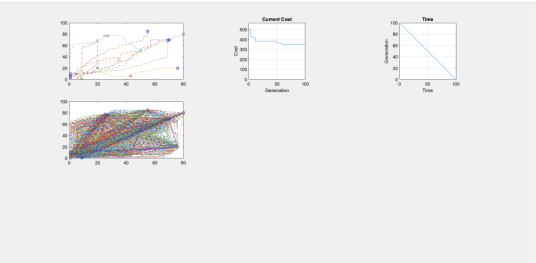

Fig. 11: Case study 1 results

As is shown in the figure above the discrete PSO was able to reach an admirable solution. When there were robots attempting to reach 15 workstations after 100 iterations the cost of the generated was 350 Case study 2 results:

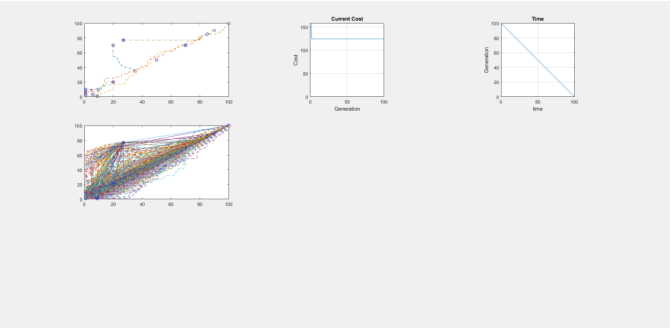

Fig. 12: Case study 2 results

When the number of workstations decreased to 10 and the number of AGVs increased to 10. We studies the behavior of the algorithm when these 2 parameters only change and the others kept constant. The best cost eas really enhanced recording a value of 125.

D. ABO results

Case study 1:

parameter	value
InitalPosition	6
workstations	15
Number of Buffalos	80
Numiterations	100
lp1	0.7
lp2	0.5

Case study 2:

parameter	value
InitalPosition	10
workstations	10
Number of Buffalos	80
Numiterations	100
lp1	0.7
lp2	0.5

The results were as follows:

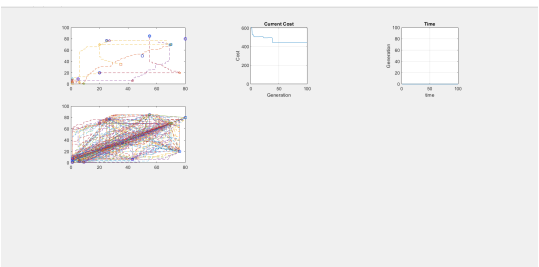

Fig. 13: Case study 1 results

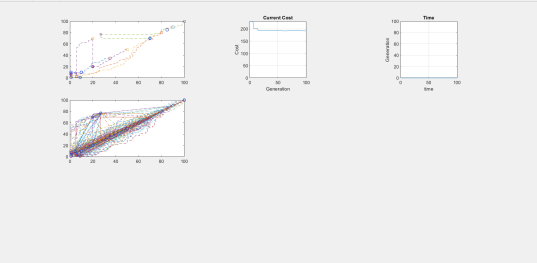

Fig. 14: Case study 2 results

The ABO yielded promising results in terms of cost function and run time. We can see that in case study 1 the results kept improving until it reaches near optimal solution. in case sudy 2 it behaved similarly to the GA which is considered the best cost function as mean and we will see that later.

E. Comparison between SA,GA,PSO,ABO

The following alogorithms were run 15 times with similar test cases. e.g: same number of workstations, same position of workstations, same number of AGVs. All similar to test case number 1 in PSO. The following results were gathered in order to apply 4 differenet performance metrics to be able to visualize and evaluate wich optimization technique gave better results between the PSO and GA and SA. The following performance metrics were:

- Computational time
- Fitness function
- standard deviation
- mean

The computation time was recorded without the visualization methods which takes up a lot of unnecessary time.

comparison	GA	SA	DiscretePSO	ABO
Time(min)(1)	1.40	0.036	1.49	2.47
Time(min)(2)	1.37	0.036	1.62	2.24
Time(min)(3)	1.37	0.035	1.59	1.96
Time(min)(4)	1.40	0.037	1.54	1.86
Time(min)(5)	1.41	0.033	1.52	1.88
Time(min)(6)	1.38	0.03	1.56	2.04
Time(min)(7)	1.41	0.032	1.58	2.22
Time(min)(8)	1.40	0.032	1.55	2.01
Time(min)(9)	1.40	0.034	1.53	1.91
Time(min)(10)	1.40	0.033	1.59	1.73
Time(min)(11)	1.44	0.033	1.56	1.74
Time(min)(12)	1.42	0.031	1.55	1.98
Time(min)(13)	1.37	0.031	1.66	2.21
Time(min)(14)	1.41	0.031	1.56	2.15
Time(min)(15)	1.40	0.036	1.55	1.86
Mean	1.40	0.033	1.56	2.02

comparison	GA	SA	DiscretePSO	ABO
Fitness(1)	369	335	383	348
Fitness(2)	350	497	408	382
Fitness(3)	306	380	301	333
Fitness(4)	393	379	335	398
Fitness(5)	357	410	288	403
Fitness(6)	369	427	412	370
Fitness(7)	273	433	383	290
Fitness(8)	299	366	390	308
Fitness(9)	336	366	324	384
Fitness(10)	349	451	388	284
Fitness(11)	352	386	329	364
Fitness(12)	327	449	410	363
Fitness(13)	330	334	422	291
Fitness(14)	275	357	371	344
Fitness(15)	282	410	426	358
Mean	331.13	398.66	371.33	348
Standard Deviation	36.96	46.43	44.87	39.27

V. Conclusion

The results recorded were as follows:

- Computational time was the best at the SA algorithm recording a mean value of 0.033445052
- The standard deviation of the GA was the best which means that every run did not deviate much from the others which shows consistency in the algorithm
- The Fitness function was the best at GA followed by the ABO then DiscretePSO then SA

VI. Future Recommendations

For future work we recommend running the techniques on more power full hardware to be able to get better results..Also the next step would be improving the system to be able to adapt to obstacles.Then implementing the GA onto a server responsible for directing the agv's to simulate in real-time the speed and accuracy of the system.

References

[1] N. Mohamad, M. H. F. Bin Md Fauadi, A. Z. Mohamed Noor, F. Jafar, and M. Ali, "Optimization of material transportation assignment for automated guided vehicle (agv) system," *International Journal of Engineering and Technology*, vol. 7, pp. 334–338, 09 2018.

[2] X. Wu, M.-X. Zhang, and Y.-J. Zheng, "An intelligent algorithm for agv scheduling in intelligent warehouses," in *International Conference on Swarm Intelligence*. Springer, 2021, pp. 163–173.

[3] X. Cao and M. Zhu, "Research on global optimization method for multiple agv collision avoidance in hybrid path," *Optimal Control Applications and Methods*, vol. 42, no. 4, pp. 1064–1080, 2021.

[4] T. Qiuyun, S. Hongyan, G. Hengwei, and W. Ping, "Improved particle swarm optimization algorithm for agv path planning," *IEEE Access*, vol. 9, pp. 33 522–33 531, 2021.