

# A FUZZY HYBRID GA-PSO ALGORITHM FOR MULTI-OBJECTIVE AGV SCHEDULING IN FMS

Mousavi, M.; Yap, H. J.<sup>#</sup>; Musa, S. N. & Dawal, S. Z. M.

Department of Mechanical Engineering, Faculty of Engineering, University of Malaya,  
50603 Kuala Lumpur, Malaysia

E-Mail: hjyap737@um.edu.my (<sup>#</sup> Corresponding author)

## Abstract

An automated guided vehicle (AGV) is a mobile robot with remarkable industrial applicability for transporting materials within a manufacturing facility or a warehouse. AGV scheduling refers to the process of allocating AGVs to tasks, taking into account the cost and time of operations. Multi-objective scheduling is adopted in this study to acquire a more complex and combinatorial model in contrast with single objective practices. The model objectives are the makespan and number of AGVs minimization while considering the AGVs battery charge. A fuzzy hybrid GA-PSO (genetic algorithm – particle swarm optimization) algorithm was developed to optimize the model. Results have been compared with GA, PSO, and hybrid GA-PSO algorithms to explore the applicability of the algorithm developed. Model's feasibility and the algorithms' performance were investigated through a numerical example before and after the optimization. The model evaluation and validation was conducted through simulation via Flexsim software. The fuzzy hybrid GA-PSO surpassed the other methods, although obtaining less mean computational time was the only significant improvement over hybrid GA-PSO.

(Received in April 2016, accepted in October 2016. This paper was with the authors 2 months for 3 revisions.)

**Key Words:** Automated Guided Vehicle, Scheduling, Multi-Objective Optimization, Genetic Algorithm, Particle Swarm Optimization, Fuzzy Hybrid GA-PSO

## 1. INTRODUCTION

AGVs are driverless mobile vehicles that are computer-controlled (usually battery operated) and equipped with different guidance systems (optical, magnetic, laser, etc.) for automated functionality [1]. AGVs are extensively used in flexible manufacturing system (FMS) for applications where long-distance horizontal transport of materials from/to multiple destination points is required and/or the material transport entails repetitive/predictable and/or dangerous tasks. Efficient scheduling of them would increase the productivity and reduce the delivery cost whilst the entire fleet is optimally utilized [2].

AGV scheduling refers to the process of allocating AGVs to tasks, taking into account the cost and required time for the operations [3, 4]. Although AGVs scheduling problem has been dealt with before [5-9], it is still an open area of research to improve it for real environment results by considering number of AGVs and their battery charge while minimizing the makespan. Makespan minimization keeps the resources utilization rate at a balanced level and results in a better implementation of expensive FMSs [8, 10]. In addition, performance of the AGV systems is heavily influenced by the number of vehicles employed, because AGVs are expensive devices that determining the type and the appropriate number of them in an FMS largely influences the profitability of the FMS [9, 11, 12], and the appropriate use of AGV's battery charge [13, 14]. Multi-objective scheduling of AGVs problem is NP hard; thus, a fuzzy hybrid GA-PSO that is a hybrid evolutionary algorithm has been applied to the proposed model in this study. GA and PSO are two well-known metaheuristic methods in optimization and both have remarkable capabilities such as 'balancing between exploration and exploitation' and 'combinatorial problem solving'. Hence, by integrating the advantages of the compensatory properties of PSO and GA their hybrid can yield better results [15-17].

However, obtaining the best performing settings of the GA operators – crossover and mutation – is a challenging task to master. The GA performance is highly dependent on the operators' value. The operators' value is usually obtained through two methods. In the first approach, the GA parameters are acquired through 'trial and error' before executing the GA, and then GA would be run using those settings being fixed. The second method applies the parameter control approach that adaptively regulates the operators' value based on the problem in question. Application of the fuzzy logic is one way of applying the second approach to the GA. This study applies a 'fuzzy automatic GA operators' controller technique (FAOCT)' to automatically adjust the GA operators during the optimization process based on the information from the previous generations such as average fitness of the population. Adaptive regulation of crossover and mutation rates is particularly attractive for its processing speed and time to optimum result compared with those using fixed values [18-21].

Overall, this research aims to use a fuzzy hybrid GA-PSO for multi-objective AGV scheduling in an FMS environment. The model will be optimized using fuzzy hybrid GA-PSO and compared to three other algorithms (GA, PSO, hybrid GA-PSO), and validated through benchmarking and simulation in Flexsim software.

## 2. PROBLEM DESCRIPTIONS AND ASSUMPTIONS

The mathematical model developed for AGV scheduling is described in detail and assumptions are introduced in this section. The three selected criteria are classified into two main objectives of (1) makespan minimization and (2) minimization of the number of AGVs while the AGVs' battery charge is taken into consideration. The assumptions and limitations applied for model development are as follows: (1) all AGVs have unit-load capacity; (2) AGVs and machines operate continuously without breakdown; (3) traffic problems, collision, or conflicts are avoided by hardware and are not considered in this study; (4) AGV loading and unloading times are considered in the travel times; (5) AGVs are allowed to park at their pick-up/drop-off (P/D) locations; (6) AGVs have a constant speed and move forward only; (7) to avoid machine deadlock, output buffers are allocated for the machines; (8) each machine operates only one product at a time; (9) the AGVs are stored in the home until dispatching commands are allocated.

### 2.1 Minimizing the makespan

Calculation of the makespan ( $MS$ ), completion time of all operations, is described in this section. A set of  $n$  jobs denoted by  $J_{jj}$  has some operations denoted by  $O_{ji}$  (Operation  $i$  from job  $j$ ), which will be processed on  $M_{ji}$  which is assigned machine number for  $O_{ji}$ . A general schematic for reading data is shown in Table I. Makespan can be expressed by:

$$MS = \max \left\{ (tDT_{ji}^a + p_{ji}) \right\} \left| \begin{array}{l} T^a = \{T_{ji}^a \mid A^a \text{ assigned to do } T_{ji}\} \\ a = 1, \dots, z \quad j = 1, \dots, n \quad i = 1, \dots, m_j \end{array} \right. \quad (1)$$

$$tDT_{ji}^a = tCA^a + UT_{ji}^a + IT_{ji}^a + LT_{ji}^a \quad (2)$$

$$\begin{aligned}
 UT_{ji}^a = tPT_{ji}^a - tCA^a = & \begin{cases} tHT_{ji}^a - tM_{j(i-1)} & \text{if } CA^a = M_{j(i-1)} \wedge i = 1 \\ tHT_{ji}^a - tM_{ji} & \text{if } CA^a = M_{ji} \wedge i = 1 \\ tHT_{ji}^a - tM_{j'i'} & \text{if } CA^a = M_{j'i'} \wedge i = 1 \\ 0 & \text{if } \begin{cases} CA^a = H \wedge i = 1 \\ \vee \\ CA^a = M_{j(i-1)} \wedge i \neq 1 \end{cases} \\ tM_{j(i-1)}T_{ji}^a - tH & \text{if } CA^a = H \wedge i \neq 1 \\ tM_{j(i-1)}T_{ji}^a - tM_{ji} & \text{if } CA^a = M_{ji} \wedge i \neq 1 \\ tM_{j(i-1)}T_{ji}^a - tM_{j'i'} & \text{if } CA^a = M_{j'i'} \wedge i \neq 1 \end{cases} \quad (3) \\
 IT_{ji}^a = & \begin{cases} p_{j(i-1)}^e - rPT_{ji}^a & \text{if } rPT_{ji}^a < p_{j(i-1)}^e \wedge rDT_{ji}^a \geq p_{j'i'}^e \wedge i \neq 1 \\ (p_{j(i-1)}^e - rPT_{ji}^a) + (p_{j'i'}^e - rDT_{ji}^a) & \text{if } rPT_{ji}^a < p_{j(i-1)}^e \wedge rDT_{ji}^a < p_{j'i'}^e \wedge i \neq 1 \\ p_{j'i'}^e - rDT_{ji}^a & \text{if } \begin{cases} rDT_{ji}^a < p_{j'i'}^e \wedge i = 1 \\ \vee \\ rDT_{ji}^a < p_{j'i'}^e \wedge rPT_{ji}^a \geq p_{j(i-1)}^e \wedge i \neq 1 \end{cases} \\ 0 & \text{if } \begin{cases} rDT_{ji}^a \geq p_{j'i'}^e \wedge i = 1 \\ \vee \\ rDT_{ji}^a \geq p_{j'i'}^e \wedge rPT_{ji}^a \geq p_{j(i-1)}^e \wedge i \neq 1 \end{cases} \end{cases} \quad (4) \\
 \text{Subject to } & M_{ji} \triangleq M_{j'i'}
 \end{aligned}$$

$$LT_{ji}^a = tDT_{ji}^a - tPT_{ji}^a = \begin{cases} tM_{ji}T_{ji}^a - tHT_{ji}^a & \text{if } PT_{ji}^a = H \wedge i = 1 \\ tM_{ji}T_{ji}^a - tM_{j(i-1)}T_{ji}^a & \text{if } PT_{ji}^a = M_{j(i-1)} \wedge i \neq 1 \end{cases} \quad (5)$$

where  $T_{ji}$  is the related task to  $O_{ji}$  (moving from  $M_{j(i-1)}$  to  $M_{ji}$  or  $H$  to  $M_{ji}$ ),  $UT_{ji}^a$  is the unloaded time of  $A^a$  doing  $T_{ji}$ ,  $LT_{ji}^a$  is the loaded time of  $A^a$  doing  $T_{ji}$ ,  $IT_{ji}^a$  is the idle time of  $A^a$  doing  $T_{ji}$ ,  $rDT_{ji}^a$  is the time that  $A^a$  reaches drop off place of  $T_{ji}$ ,  $tDT_{ji}^a$  is the drop off time of  $A^a$  doing  $T_{ji}$  and  $p_{ji}$  is the processing time of  $O_{ji}$ .  $tPT_{ji}^a$  is pick-up time of  $A^a$  doing  $T_{ji}$ ,  $CA^a$  is the current position of  $A^a$ ,  $tCA^a$  is the time of current position of  $A^a$ , and  $H$  is the load/unloading (Home) point.

Subject to:

$$CTO_{ji} \geq p_{ji}^s \quad \forall i = 1 \quad (6)$$

$$tPT_{ji}^a \geq 0 \quad \forall T_{ji}^a \in T^a \quad (7)$$

$$p_{ji}^s \geq tPT_{j(i+1)}^a - p_{ji} \quad \forall j, i \quad (8)$$

$$p_{ji}^s - p_{j(i-1)}^s \geq p_{ji} + LT_{ji}^a \quad \forall j, i = 2, \dots, m_j \quad (9)$$

$$\begin{aligned}
 (p_{ji}^s - p_{j'i'}^s - p_{j'i'} + \mu |M_{ji} - M_{j'i'}| \geq 0) \quad \vee \\
 (p_{j'i'}^s - p_{ji}^s - p_{ji} + \mu |M_{ji} - M_{j'i'}| \geq 0) \quad \forall (j, i, j', i') \quad (10)
 \end{aligned}$$

$$\begin{aligned}
 (tPT_{jm_j}^a - tPT_{j'i'}^a - LT_{j'i'}^a + \mu |T_{ji}^a - T_{j'i'}^a| \geq 0) \quad \vee \\
 (tPT_{j'i'}^a - tPT_{jm_j}^a - LT_{ji}^a + \mu |T_{ji}^a - T_{j'i'}^a| \geq 0) \quad \forall (j, m_j, j', i') \quad (11)
 \end{aligned}$$

where  $m_{j,j'}$  is the total number of operations of job  $j,j'$ ,  $i,i'$  are the indexes of operations,  $i,i' = 1, 2, \dots, m_{j,j'}$ ,  $n$  is total number of jobs,  $LT_{ji}^a$  is the loaded time of  $A^a$  doing  $T_{ji}$ ,  $p_{ji}$  is the processing time of  $O_{ji}$ ,  $p_{ji}^s$  is the start time of processing time of  $O_{ji}$ ,  $p_{ji}^e$  is the end time of processing time of  $O_{ji}$  and  $T_{ji}^a$  is the assigned  $A^a$  to do task  $T_{ji}$ .  $CTO_{ji}$  is the completion time of operation  $O_{ji}$ ,  $\mu$  is a large positive number,  $a,a'$  is index of AGVs,  $(a,a' = 1, \dots, z)$ , and  $A^{a,a'}$  represents AGVs.

Constraint (6) is applied to ensure that completion time of the first operation of each job is feasible. Operations precedence constraint is described through inequality (8). The (10) and (11) inequalities respectively denote the operation and the AGV un-overlapping constraints.

## 2.2 Minimizing the number of AGVs

This section describes the procedure for calculating the number of AGVs, denoted by ( $NA$ ), while taking into account the sufficiency of the AGVs' battery charge.  $T_{ji}$  is the related task to  $O_{ji}$  (moving from  $M_{ji-1}$  to  $M_{ji}$  or  $H$  to  $M_{ji}$ ),  $T_{ji}^a$  is the assigned  $A^a$  to do task  $T_{ji}$ .  $T^a$  is a collection of operations that have done by  $A^a$ .  $ChHT_{ji}^a$  is the charge that  $A^a$  needed for doing the task  $T_{ji}$  and return home.  $ChA^a$  is the current battery charge of  $A^a$ .  $tCPT_{ji}^a$  is the travel time of  $A^a$  from its current point to reach the start point of  $T_{ji}$ .  $tDT_{ji}^a$  is the drop off time of  $A^a$  doing  $T_{ji}$ .  $tT_{ji}^a H$  is the time that  $A^a$  arrives home after doing  $T_{ji}$ . Number of AGV can be expressed by:

$$NA = \max \{a\} \mid T = \{T^a\} \quad \wedge \quad (12)$$

$$T^a = \{T_{ji}^a \mid A^a \text{ assigned to do } T_{ji}\} \quad a = 1, \dots, z \quad j = 1, \dots, n \quad i = 1, \dots, m_j$$

$$A^a \text{ is assigned to } T_{ji} \text{ (to create } T_{ji}^a) \text{ if } \begin{cases} ChA^a \geq ChHT_{ji}^a \\ \wedge \\ tCPT_{ji}^a < tCPT_{ji}^{a'} \\ \vee \\ tCPT_{ji}^a = tCPT_{ji}^{a'} \quad \wedge \quad a < a' \end{cases} \quad (13)$$

$$ChHT_{ji}^a = \gamma(UT_{ji}^a + LT_{ji}^a + (tT_{ji}^a H - tDT_{ji}^a)) \quad (14)$$

$$A^y \mid y \text{ is a new AGV} \text{ is assigned to } T_{ji} \text{ (to create } T_{ji}^y) \text{ if } \{tCPT_{ji}^y + RT_{ji}^y \leq \lambda(tCPT_{ji}^a + RT_{ji}^a)\} \quad (15)$$

where  $RT_{ji}^a$  is the running time (loaded + unloaded) of  $A^a$  doing  $T_{ji}$ ,  $\lambda$  is a coefficient for determining when a new AGV should be added,  $\gamma$  is the ratio of energy consumption to time,  $ChA^a$  is the current battery charge of  $A^a$ ,  $ChHT_{ji}^a$  is the charge that  $A^a$  needed for doing the task  $T_{ji}$  and back home,  $tT_{ji}^a H$  is the time that  $A^a$  arrives home after doing  $T_{ji}$ ,  $PT_{ji}^a$  is the pick-up point of  $A^a$  doing  $T_{ji}$ ,  $tPT_{ji}^a$  is the pick-up time of  $A^a$  doing  $T_{ji}$ ,  $tDT_{ji}^a$  is the drop off time of  $A^a$  doing  $T_{ji}$ ,  $rPT_{ji}^a$  is the time that  $A^a$  reaches pick-up place of  $T_{ji}$ ,  $y$  is the index of new AGVs.

Eq. (13) makes sure that the assigned AGV has enough battery charge to do the job and return home, while it chooses the AGV, which takes less time to reach the point. As battery run time of an AGV and battery charging time can be defined depending on the type of batteries used/charging methods, charge rate, application, manufacturer, and assignments the vehicles perform,  $\lambda$  has been defined to adapt to any kind of battery, method charge, etc. The automatic and opportunity battery charging is considered here and, on average, an AGV charges for 10-12 minutes every hour in this method [22, 23].

### 2.3 Multi-objective evaluation

Choosing a solution out of all the efficient solutions is referred to as a posteriori approach by decision makers. Pareto is a prominent approach for optimizing the multi-objective problems. In this method, Pareto-optimal set is a group of best trade-off schedules, and Pareto-front refers to a set of Pareto solutions [19]. Overall fitness function ( $f(x)$ ) formulation is described by:

$$f(x) = \sum_{\beta=1}^{\beta=L} g_{\beta} w_{\beta} f_{\beta}(x) \quad (16)$$

where  $w_{\beta}$  is the weight of  $\beta^{th}$  objective function,  $g_{\beta}$  is the coefficient for  $\beta^{th}$  objective function to obtain identical ranges of value among the objectives [24, 25], and  $\sum_{\beta=1}^{\beta=L} w_{\beta} = 1$ .

In this study, different weights are applied to the two main objectives as  $\frac{1}{3}$  for AGV numbers and  $\frac{2}{3}$  for makespan. To have a similar variation range in both the objectives, a coefficient is applied to the AGV number objective. The overall fitness function is calculated by:

$$f(x) = \frac{2}{3}(MS) + \frac{1}{3}(g_{\beta})(NA) \quad (17)$$

## 3. PROPOSED ALGORITHM

### 3.1 Fuzzy hybrid GA-PSO

GA and PSO are two well-known metaheuristic methods in optimization. PSO is a robust algorithm which its remarkable performance has been proved in many researches [26]. One of the most significant advantages of PSO application is the simplicity in understanding and implementation. Besides, balancing between exploration and exploitation in the PSO can be taken care by means of three parameters, which can be adaptive or constant during the run. GA has the capability of simultaneous evaluation of many points in the search area, which increases the probability of finding the global solution of the problem [27-30]. GA and PSO methods for the model have been explained in [17]. By hybridizing GA and PSO, the natural capabilities (balancing between exploration and exploitation and combinatorial problem solving) of these search methods can promise better performance in the problem involved [17].

Fuzzy hybrid GA-PSO is a method which got some of the PSO parameters and some of the GA operators in fuzzy mode to improve the quality of results. The method is illustrated in Fig. 1.

Similar to all optimization algorithms, this fuzzy hybrid algorithm starts with the initializing of the parameters and particles. Application parameters and conditions are extracted from Table I. The first column shows a particle ( $PR_{\alpha}$ ) and the second one shows dimensions of the particle ( $d$ ),  $d = 1, \dots, \theta$ , and  $\theta$  is the total number of operations. The dimensions' codes are presented in the first column, which will be discussed later. The dimension number of each particle is the total number of operations to be done in the scheduling. Initializing particles population is done with randomizing their position ( $Q$ ) and velocity ( $V$ ) within the maximum and minimum limits which are  $q_{min} = 0$ ,  $q_{max} = 0$ , and  $v_{min} = 0$ ,  $v_{max} = 0$ . A particle is every potential sequence of operations, and each operation is represented by a particle dimension. Three sub-steps for encoding a particle are as follows (for more details refer to [17]):

1) Applying smallest position value (SPV) rule. SPV is a rule that facilitates transformation of the continuous PSO algorithm to discrete cases applicable to all types of the scheduling problem [31].

2) Assigning the dimensions' code to the particles. Dimensions' codes, as in the column three of the Table I, are given to the related particles. Dimensions' codes are the same as the job number.

3) Identifying the operations sequence in each job. From the left side, the first presence of a job number in the sequence is presumed the first operation of that job (i.e.,  $O_{j1}$ ). In a similar fashion, the second time appearance of the same job number in the sequence defines the second operation of the same job (i.e.,  $O_{j2}$ ) and so on. Upon assigning the first encountered generated number to a job's first operation, the precedence constraints are then automatically operated in this technique.

The final encoded solution should undertake the evaluation process. This problem has two main objectives, minimizing the makespan and number of AGVs as in Eqs. (1) to (15). Next, Eq. (17) computes the total fitness values of the efficient frontiers. After evaluating the solution, personal best  $B_{ad}^t$  and global best  $G_d^t$  are stored in the memory. Personal best is the best position of  $\alpha^{\text{th}}$  particle on  $d^{\text{th}}$  dimension found so far and the global best is the best particle of the whole swarm on  $d^{\text{th}}$  dimension found so far.

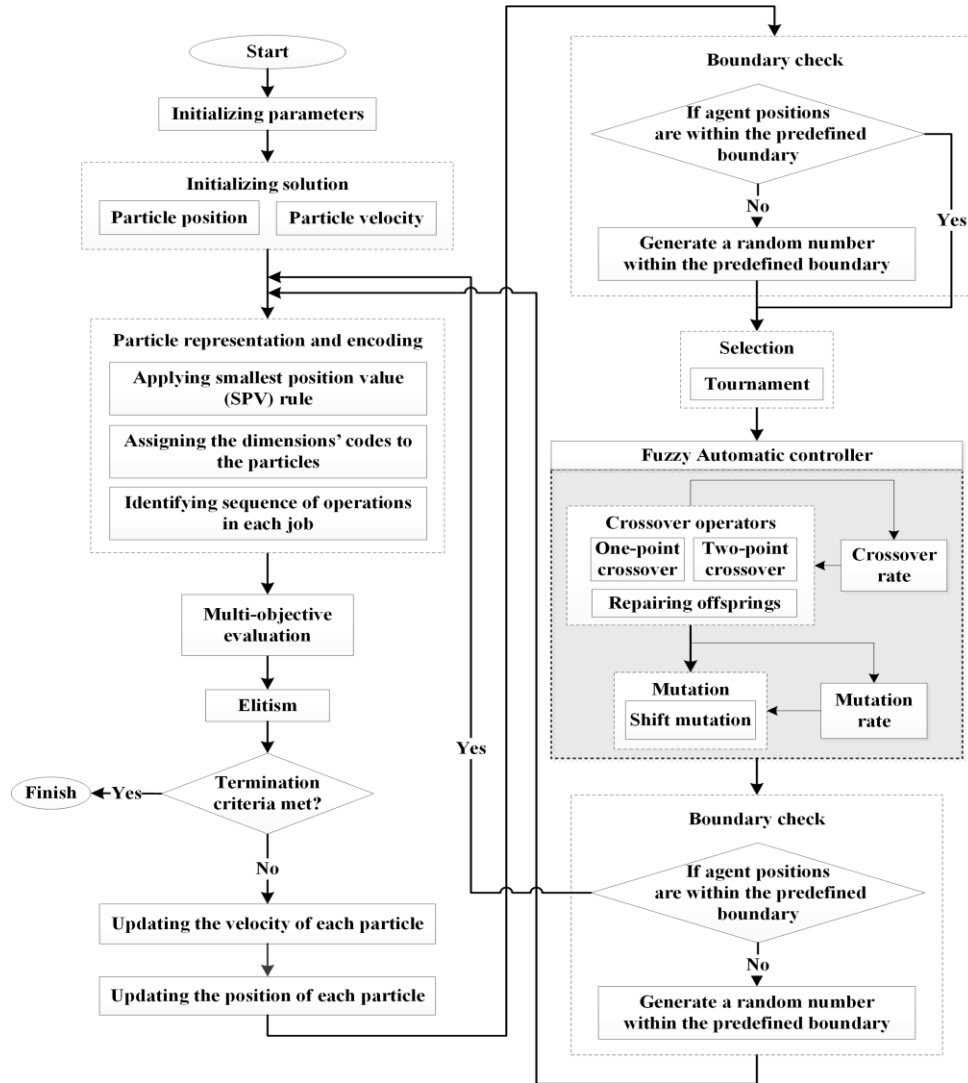


Figure 1: Flowchart of fuzzy hybrid GA-PSO.

In PSO method, each iteration consists of updating particles based on the global and local best particles from the previous iteration. The new velocity and position of the particle is calculated based on the Eqs. (18) and (19).

Table I: General schematic for reading data.

Particle ( $PR_a$ )	Dimension number ( $d$ )	Dimension code ( $j$ )	Job ( $J_i$ )	Operation ( $O_{ij}$ )	Machine ( $M_{ij}$ )	Processing time ( $p_{ij}$ )
$PR_1$	1	1	$J_1$	$O_{11}$	$M_{11}$	$p_{11}$
	2	1		$O_{12}$	$M_{12}$	$p_{12}$
	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$
	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$
	$m_1$	1		$O_{1m_1}$	$M_{1m_1}$	$p_{1m_1}$
	$\vdots$	2	$J_2$	$O_{21}$	$M_{21}$	$p_{21}$
	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$
	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$
	$\vdots$	2		$O_{2m_2}$	$M_{2m_2}$	$p_{2m_2}$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$\theta$	$n$	$J_n$	$O_{n1}$	$M_{n1}$	$p_{n1}$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$\theta$	$n$		$O_{nm_n}$	$M_{nm_n}$	$p_{nm_n}$

$$v_{\alpha d}^{t+1} = \omega v_{\alpha d}^t + C_1 \phi_1 (B_{\alpha d}^t - q_{\alpha d}^t) + C_2 \phi_2 (G_d^t - q_{\alpha d}^t) \quad (18)$$

$$q_{\alpha d}^{t+1} = q_{\alpha d}^t + v_{\alpha d}^{t+1} \quad (19)$$

where  $v_{\alpha d}^t$  and  $v_{\alpha d}^{t+1}$  are the velocity of  $\alpha^{\text{th}}$  particle on  $d^{\text{th}}$  dimension at instance ( $t$ ) and ( $t+1$ ), respectively;  $q_{\alpha d}^t$  and  $q_{\alpha d}^{t+1}$  are the position of  $\alpha^{\text{th}}$  particle on  $d^{\text{th}}$  dimension at instance ( $t$ ) and ( $t+1$ ), respectively.  $t$  is the previous iteration,  $d$  is the dimension and  $\alpha$  is the index for particles,  $\alpha=1, \dots, S^t$ ,  $S^t$  is swarm size at iteration ( $t$ ),  $\phi_1$  and  $\phi_2$  are uniformly distributed random numbers in the interval of  $[0, 1]$ .  $C_1$  is self-confidence while  $C_2$  is swarm confidence and their values should be tuned based on the experiment.  $\omega$  is the inertia weight parameter. This parameter is the level which previous velocity effects on the current velocity [32, 33]. Inertia weight is decayed in related to the number of iteration, as shown in Eq. (20).

$$\omega = \omega_{\max} - It \times \frac{\omega_{\max} - \omega_{\min}}{It_{\max}} \quad (20)$$

where  $It$  is the current iteration number, and  $It_{\max}$  denotes the maximum number of iteration,  $\omega_{\max}$  and  $\omega_{\min}$  are maximum and minimum values of  $\omega$  respectively.

After updating the particles, the two operators derived from GA, which are crossover and mutation, are used for the next generation. The crossover operator produces two new chromosomes by exchanging some genes of the two selected chromosomes. The GA mutation operator initiates extra variability to maintain the diversity within a population. Mutation operator is not applied on immune chromosomes (more details in [17]).

### 3.2 Adaptive genetic operators (AGOs)

GA parameters such as adaptive crossover and mutation probabilistic rates with fuzzy logic rule-based will greatly assist in identifying optimum results quickly compared with those using fixed values [20]. Adaptive regulation of crossover and mutation rates is particularly attractive to achieve optimum results and has been implemented in a number of works [20, 21, 34].

The number of crossovers is calculated using:

$$\text{Number of crossovers} = \frac{(CR(t) \times PS)}{2} \quad (21)$$

where  $CR(t)$  is crossover rate (will be explained below) and  $PS$  is population size.

The number of mutations is computed by:

$$\text{Number of mutations} \cong (PS \times \text{Max}.G_e) \times Pm(t) \quad (22)$$

where  $Pm(t)$  is mutation rate (will be explained below), and  $\text{Max}.G_e$  is maximum gene code.

### 3.3 Fuzzy automatic GA operators' controller technique (FAOCT)

FAOCT is applied to automatically calibrate the GA operators of crossover and mutation during the optimization (Fig. 2).

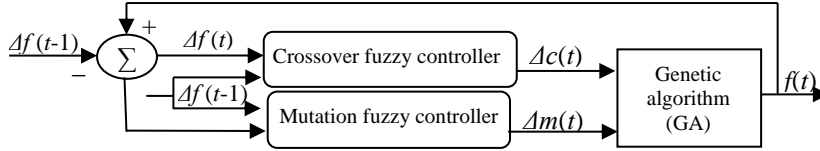


Figure 2: Block diagram of proposed fuzzy automatic GA operators' controller technique for crossover and mutation rates.

$\Delta c(t)$  and  $\Delta m(t)$  shown in Fig. 2 are the average crossover rate and average mutation rate based on generation ( $t$ ), respectively, and  $\Delta f(t)$  is the average fitness based on generation ( $t$ ). The heuristic updating strategy for these two operators is based on the changes in the average fitness of the GA population for two continuous generations. The crossover rate, mutation rate, and occurrence rates of the operators will increase if they consistently produce better offspring during the recombination process. Otherwise, rate of the operators will decrease [34, 35]. The changes in fitness at two consecutive steps (i.e.  $\Delta f(t-1)$  and  $\Delta f(t)$ ) are inputs of the fuzzy crossover and mutation rates, while the changes in the crossover and mutation rates are the outputs.  $\Delta f(t)$  is set using Eq. (23):

$$\Delta f(t) = \left( \overline{f_{par-size}}(t) - \overline{f_{off-size}}(t) \right) \times \delta = \left( \frac{\sum_{k=1}^{par-size} f_k(t)}{par-size} - \frac{\sum_{k=par-size+1}^{par-size+off-size} f_k(t)}{off-size} \right) \times \delta \quad (23)$$

where  $\delta$  is a scaling factor to normalize the average fitness values for defuzzification in the fuzzy logic controller (FLC), and is varied accordingly to the given problem.  $\overline{f_{par-size}}(t)$  and  $\overline{f_{off-size}}(t)$  are the average fitness values of parents, and offspring at generation ( $t$ ), respectively. The regulation and procedure of  $\Delta c(t)$ , and  $\Delta m(t)$  begin with the application of  $\Delta f(t-1)$  and  $\Delta f(t)$  based on the average fitness values, as follows:

If  $\varepsilon \leq \Delta f(t-1) \leq \gamma$  and  $\varepsilon \leq \Delta f(t) \leq \gamma$ , then increase  $CR(t)$ , and  $Pm(t)$  for next generation;  
 If  $-\gamma \leq \Delta f(t-1) \leq -\varepsilon$  and  $-\gamma \leq \Delta f(t) \leq -\varepsilon$ , then decrease  $CR(t)$ , and  $Pm(t)$  for next generation;  
 If  $-\varepsilon \leq \Delta f(t-1) \leq \varepsilon$  and  $-\varepsilon \leq \Delta f(t) \leq \varepsilon$ , then rapidly increase  $CR(t)$ , and  $Pm(t)$  for next generation;  
 where  $\varepsilon$  is a given real number in the proximity of zero,  $\gamma$  and  $-\gamma$  represent the given maximum and minimum value, respectively, for the fuzzy membership function of the fuzzy input and output linguistic variables. The  $\Delta f(t-1)$  and  $\Delta f(t)$  values are normalized correspondingly within the range of  $[-1.0, 1.0]$ . The  $\Delta c(t)$ , and  $\Delta m(t)$  values are normalized within the range of  $[-0.1, 0.1]$ , and  $[-0.01, 0.01]$ , respectively, depending on their corresponding maximum values. The application of a fuzzy decision table for crossover and mutation rates is given in Table II. The labels of the membership function are as follows: PL = Positive larger, PG = Positive large, PM = Positive medium, PS = Positive small, NL = Negative larger, NG = Negative large, NM = Negative medium, NS = Negative small, ZE = Zero. The quantified levels corresponding to the linguistic values of input and output fuzzy variables of crossover and mutation rates FLCs are designated as  $-4, -3, -2, -1, 0, 1, 2, 3$ , and  $4$ , respectively and are shown in Table III. Fuzzy inference engine uses the rules of the fuzzy rule base to generate fuzzy outputs based on the inputs. The input values are assigned to the indices  $x$  and  $y$  which represent the first and second inputs of the average fitness functions  $\Delta f(t-1)$  and  $\Delta f(t)$ , respectively. The outputs of changes in the FLC for crossover and mutation



rates are generated upon identification of  $Z(x,y)$  for crossover and mutation rates using Table III and Eqs. (24) and (25), respectively:

$$\Delta c(t) = Z(x, y) \times 0.02 \quad (24)$$

$$\Delta m(t) = Z(x, y) \times 0.002 \quad (25)$$

$$\text{Subject to} \quad Z \leq vx + (1 - v)y \quad (26)$$

where  $Z(x,y)$  consists of the corresponding values of  $\Delta f(t-1)$  and  $\Delta f(t)$  for defuzzification,  $x, y \in \{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$ , and  $v$  is an adaptive coefficient which varies with the fitness value of the whole population. Based on trial and error, the crossover and mutation rates FLCs has a better performance when  $v = 0.5$ . To regulate the increasing and decreasing range of rates, the value 0.02 is selected for crossover operator and 0.002 for the mutation operator.

Table II: Fuzzy decision table for crossover and mutation rate.

$\Delta c(t) \& \Delta m(t)$		$\Delta f(t-1)$									
		NL	NG	NM	NS	ZE	PS	PM	PG	PL	
$\Delta f(t)$	NL	NL	NG	NG	NM	NM	NS	NS	ZE	ZE	
	NG	NG	NG	NM	NM	NS	NS	ZE	ZE	PS	
	NM	NG	NM	NM	NS	NS	ZE	ZE	PS	PS	
	NS	NM	NM	NS	NS	ZE	ZE	PS	PS	PM	
	ZE	NM	NS	NS	ZE	ZE	PS	PS	PM	PM	
	PS	NS	NS	ZE	ZE	PS	PS	PM	PM	PG	
	PM	NS	ZE	ZE	PS	PS	PM	PM	PG	PG	
	PG	ZE	ZE	PS	PS	PM	PM	PG	PG	PL	
	PL	ZE	PS	PS	PM	PM	PG	PG	PL	PL	

Table III: Look-up table for control actions by crossover rate, and mutation rate FLC.

$Z(x,y)$		$x$									
		-4	-3	-2	-1	0	1	2	3	4	
$y$	-4	-4	-3	-3	-2	-2	-1	-1	-0	+0	
	-3	-3	-3	-2	-2	-1	-1	-0	+0	1	
	-2	-3	-2	-2	-1	-1	-0	+0	1	1	
	-1	-2	-2	-1	-1	-0	+0	1	1	2	
	0	-2	-1	-1	-0	+0	1	1	2	2	
	1	-1	-1	-0	+0	1	1	2	2	3	
	2	-1	-0	+0	1	1	2	2	3	3	
	3	-0	+0	1	1	2	2	3	3	4	
	4	+0	1	1	2	2	3	3	4	4	

The crossover and mutation rates are respectively updated by the below formulas,

$$CR(t) = CR(t-1) + \Delta c(t) \quad (27)$$

$$Pm(t) = Pm(t-1) + \Delta m(t) \quad (28)$$

The calibrated rates of  $CR(t)$  and  $Pm(t)$  should not exceed the range of 0.5 to 1.0 and 0.0 to 0.1, respectively [35].

This procedure would be repeated until reaching the maximum number of iteration. The particle with global best is returned as the best solution once the termination criteria are met.

## 4. COMPUTATIONAL RESULTS AND DISCUSSION

To validate the model, a testbed with 6 jobs ( $J_1, \dots, J_6$ ) processing on 12 machines ( $M_1, \dots, M_{12}$ ), and each job with 3 to 8 operations, has been used (Fig. 3). Table IV shows the AGV travel time between H (L/U) points and machines, and Table V demonstrates the processing time of every operation on different machines.

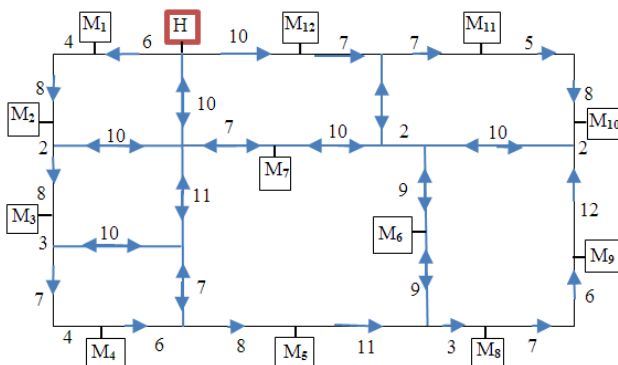


Figure 3: The example layout.

Table IV: AGV travel time between L/U points and machines.

	L/U	M <sub>1</sub>	M <sub>2</sub>	M <sub>3</sub>	M <sub>4</sub>	M <sub>5</sub>	M <sub>6</sub>	M <sub>7</sub>	M <sub>8</sub>	M <sub>9</sub>	M <sub>10</sub>	M <sub>11</sub>	M <sub>12</sub>
L/U	0	6	18	28	42	36	38	17	50	51	37	24	10
M <sub>1</sub>	34	0	12	22	36	50	52	31	64	77	71	58	44
M <sub>2</sub>	22	28	0	10	24	38	40	19	52	66	59	46	32
M <sub>3</sub>	34	40	52	0	14	28	48	31	42	55	71	58	44
M <sub>4</sub>	34	40	52	42	0	14	34	31	28	41	71	58	44
M <sub>5</sub>	58	64	76	66	80	0	20	41	14	27	61	48	68
M <sub>6</sub>	38	46	58	46	60	54	0	21	12	25	41	28	48
M <sub>7</sub>	17	23	35	25	39	33	21	0	33	46	40	27	27
M <sub>8</sub>	64	70	82	72	86	80	44	47	0	13	67	54	74
M <sub>9</sub>	51	57	69	59	73	67	31	34	43	0	54	41	61
M <sub>10</sub>	41	47	59	49	63	57	21	24	33	46	0	31	51
M <sub>11</sub>	54	60	72	62	76	70	34	37	46	59	13	0	64
M <sub>12</sub>	44	50	62	52	66	60	28	27	40	53	27	14	0

Table V: The processing time (in min) of every operation on different machines.

Code	Operation	Machine	Operation time	Code	Operation	Machine	Operation time
1	O <sub>11</sub>	M <sub>2</sub>	37	3	O <sub>35</sub>	M <sub>6</sub>	25
1	O <sub>12</sub>	M <sub>6</sub>	33	3	O <sub>36</sub>	M <sub>7</sub>	13
1	O <sub>13</sub>	M <sub>5</sub>	34	3	O <sub>37</sub>	M <sub>8</sub>	14
1	O <sub>14</sub>	M <sub>8</sub>	35	3	O <sub>38</sub>	M <sub>11</sub>	23
1	O <sub>15</sub>	M <sub>1</sub>	23	4	O <sub>41</sub>	M <sub>1</sub>	16
1	O <sub>16</sub>	M <sub>12</sub>	34	4	O <sub>42</sub>	M <sub>7</sub>	11
1	O <sub>17</sub>	M <sub>7</sub>	37	4	O <sub>43</sub>	M <sub>5</sub>	23
1	O <sub>18</sub>	M <sub>5</sub>	26	4	O <sub>44</sub>	M <sub>12</sub>	34
2	O <sub>21</sub>	M <sub>3</sub>	23	4	O <sub>45</sub>	M <sub>6</sub>	25
2	O <sub>22</sub>	M <sub>4</sub>	26	4	O <sub>46</sub>	M <sub>8</sub>	13
2	O <sub>23</sub>	M <sub>6</sub>	27	5	O <sub>51</sub>	M <sub>1</sub>	16
2	O <sub>24</sub>	M <sub>11</sub>	25	5	O <sub>52</sub>	M <sub>7</sub>	11
2	O <sub>25</sub>	M <sub>10</sub>	34	5	O <sub>53</sub>	M <sub>9</sub>	31
2	O <sub>26</sub>	M <sub>9</sub>	23	6	O <sub>61</sub>	M <sub>3</sub>	26
3	O <sub>31</sub>	M <sub>1</sub>	26	6	O <sub>62</sub>	M <sub>2</sub>	31
3	O <sub>32</sub>	M <sub>2</sub>	25	6	O <sub>63</sub>	M <sub>10</sub>	23
3	O <sub>33</sub>	M <sub>10</sub>	31	6	O <sub>64</sub>	M <sub>4</sub>	24
3	O <sub>34</sub>	M <sub>4</sub>	24	6	O <sub>65</sub>	M <sub>11</sub>	35

Fig. 4 shows the scheduling before optimization, where the makespan is 1475 minutes using 7 AGVs.

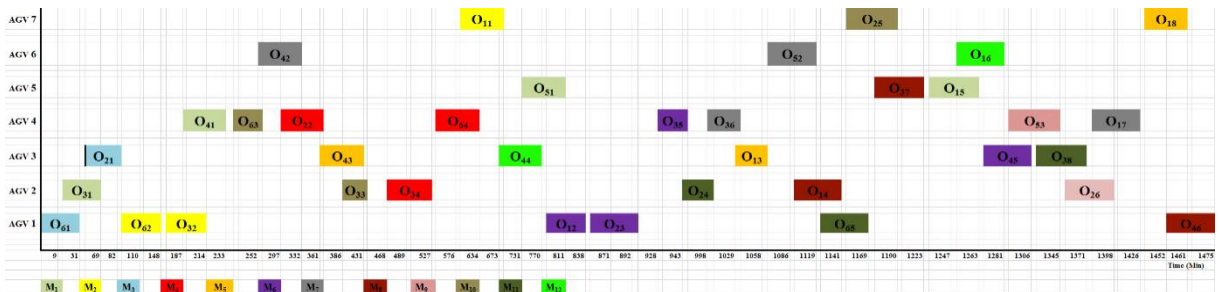


Figure 4: The example before optimization.

The best performing GA parameters set, according to the experimental approach, was found to be the crossover and mutation rates of 0.2 and 0.03, respectively. The best performing of PSO parameters were  $C_1 = C_2 = 2$ ,  $w_{min} = 0.01$ , and  $w_{max} = 0.5$ . All the four algorithms, after the optimization, showed satisfactory results in decreasing the makespan and AGVs number. Their performance is illustrated in Fig. 5.

Table VI illustrates that the fuzzy hybrid GA-PSO algorithm has outperformed the other three algorithms and showed the best fitness result by 1181.1, followed by hybrid GA-PSO (1182.4), GA (1229.1), and PSO (1317.5). However, its only notable improvement over hybrid GA-PSO was obtaining less mean computational time.

Fig. 6 shows scheduling of the example after optimization by fuzzy hybrid GA-PSO with makespan of 1229 minutes and 5 AGVs.

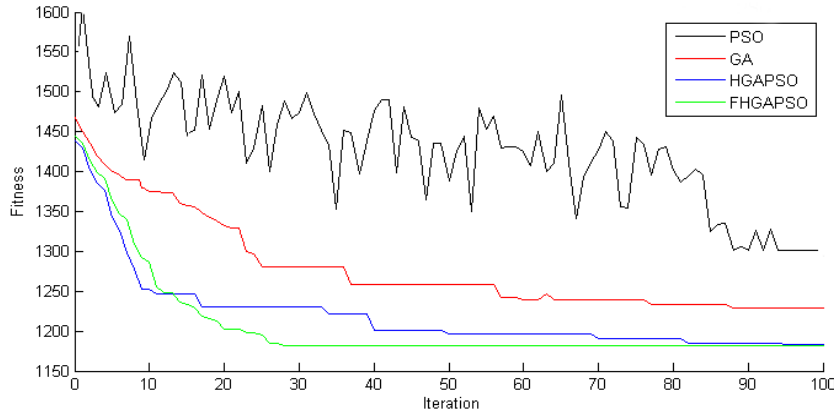


Figure 5: Performance of the four algorithms.

Table VI: Test results of the optimization algorithms.

Algorithm	Objectives	Best	Worst	Mean	Mean computational time
PSO	Fitness function	1317.5349	1368.1965	1339.3993	128.7481 sec
	Makespan	1325	1401	1357.8	
	Number of AGV	6	6	6	
GA	Fitness function	1229.1548	1352.8647	1302.4750	131.2939 sec
	Makespan	1301	1378	1345.2	
	Number of AGV	5	6	5.6	
Hybrid GA-PSO	Fitness function	1182.4928	1218.4892	1189.9587	133.6356 sec
	Makespan	1231	1285	1242.2	
	Number of AGV	5	5	5	
Fuzzy hybrid GA-PSO	Fitness function	1181.1597	1219.1558	1189.5587	105.3956 sec
	Makespan	1229	1286	1241.6	
	Number of AGV	5	5	5	

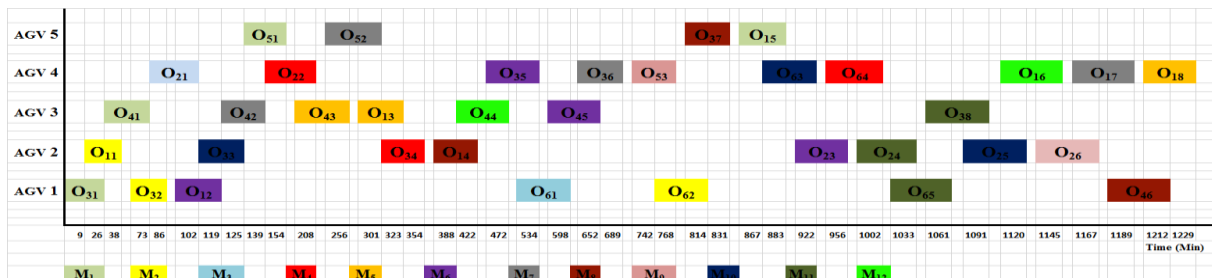


Figure 6: The example after optimization (by fuzzy hybrid GA-PSO).

## 5. SIMULATION VIA FLEXSIM

The practicality of the proposed model is investigated through simulation with Flexsim software based on the above numerical example. Fig. 7 shows a part of the model that was built based on the given layout in Fig. 3. In the simulated model, each part enters the system through 'Source' and the sequence defined in the source represents the job sequence. 'Sink'

acts as home (H), the place that all the parts are distributed from. Processors are used as machines and a 'Queue' object placed after each machine acts as a space for the processed parts waiting for the AGV pick-up. The yellow vehicles in Fig. 7 are the AGVs that collect the products either from the sink resource or machines and deliver as scheduled. Using this configuration, the makespan magnitude obtained in simulation was equal to that of the three algorithms confirming the optimization results. The validity and practicality of the proposed model is therefore proved through this experiment, in which it provides a benchmark for further studies on AGVs' scheduling. The model compatibility to different FMSs and environments was also experimented using other examples and was proved feasible. The model was also proved capable of handling the objectives either separately or in a combination.

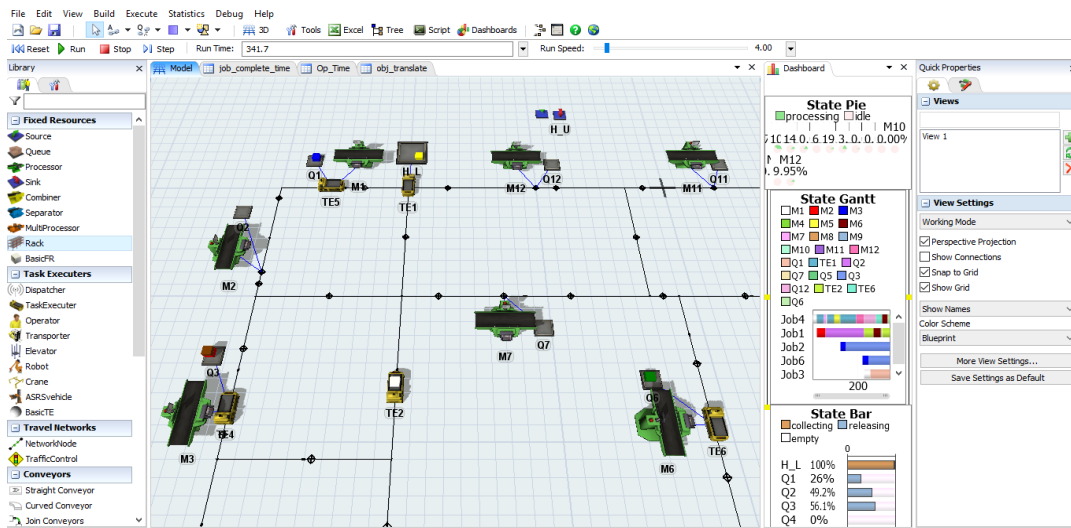


Figure 7: Simulation of the sample by Flexsim software.

## 6. CONCLUSION

The main draw of this research was to use fuzzy hybrid GA-PSO algorithm for optimization of a multi-objective AGV scheduling model in an FMS and compare the results obtained with three other algorithms (GA, PSO, and hybrid GA-PSO). The model's constructing criteria were minimization of AGVs number and makespan, while considering the battery charge of AGVs. The near-optimum schedules for the combined objective functions were obtained in the numerical examples. Comparison of the four algorithms' result revealed that the fuzzy hybrid GA-PSO yields the least makespan and AGV numbers. However, a marginal difference between fuzzy hybrid GA-PSO and hybrid GA-PSO performances was observed, in which the only significant improvement over hybrid GA-PSO was the less mean computational time obtained. The AGV system simulation using Flexsim software, similar to the experimental results, proved the developed model's practicality and the studied algorithms appropriateness for solving the scheduling problem. Final, the developed model is applicable in various FMS configurations and environments, and it is suitable for single or combinatorial optimization of the objectives set.

## ACKNOWLEDGEMENT

The authors would like to acknowledge for the University of Malaya for the financial support under UMRG Top Down Programme (Grant No. RP027-14AET). The authors also like to acknowledge the Ministry of Higher Education of Malaysia for the support under High Impact Research Grant UM.C/HIR/MOHE/ENG/35 (D000035-16001).

## REFERENCES

- [1] Vasava, A. S. (2014). Scheduling of automated guided vehicle in different flexible manufacturing system environment, *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, Vol. 1, No. 8, 262-267
- [2] Reddy, B. S. P.; Rao, C. S. P. (2011). Flexible manufacturing systems modelling and performance evaluation using Automod, *International Journal of Simulation Modelling*, Vol. 10, No. 2, 78-90, doi:[10.2507/IJSIMM10\(2\)3.176](https://doi.org/10.2507/IJSIMM10(2)3.176)
- [3] Pan, X. Y.; Wu, J.; Zhang, Q. W.; Lai, D.; Xie, H. L.; Zhang, C. (2013). A case study of AGV scheduling for production material handling, *Applied Mechanics and Materials*, Vol. 411-414, 2351-2354, doi:[10.4028/www.scientific.net/AMM.411-414.2351](https://doi.org/10.4028/www.scientific.net/AMM.411-414.2351)
- [4] Udhayakumar, P.; Kumanan, S. (2010). Task scheduling of AGV in FMS using non-traditional optimization techniques, *International Journal of Simulation Modelling*, Vol. 9, No. 1, 28-39, doi:[10.2507/IJSIMM09\(1\)3.139](https://doi.org/10.2507/IJSIMM09(1)3.139)
- [5] Aydemir, E.; Koruca, H. I. (2015). A new production scheduling module using priority-rule based genetic algorithm, *International Journal of Simulation Modelling*, Vol. 14, No. 3, 450-462, doi:[10.2507/IJSIMM14\(3\)7.299](https://doi.org/10.2507/IJSIMM14(3)7.299)
- [6] Ventura, J. A.; Pazhani, S.; Mendoza, A. (2015). Finding optimal dwell points for automated guided vehicles in general guide-path layouts, *International Journal of Production Economics*, Vol. 170, Part C, 850-861. doi:[10.1016/j.ijpe.2015.03.007](https://doi.org/10.1016/j.ijpe.2015.03.007)
- [7] Cai, Q.; Tang, D.; Zheng, K.; Zhu, H.; Wu, X.; Lu, X. (2014). Multi-AGV scheduling optimization based on neuro-endocrine coordination mechanism, *International Journal on Smart Sensing and Intelligent Systems*, Vol. 7, No. 4, 1613-1630
- [8] Huang, D.; Zhang, G. (2013). Scheduling control of AGV system based on game theory, *6<sup>th</sup> IEEE International Conference on Advanced Infocomm Technology (ICAIT)*, 223-227, doi:[10.1109/ICAIT.2013.6621564](https://doi.org/10.1109/ICAIT.2013.6621564)
- [9] Wang, J. B.; Hou, L. Y.; Li, W.; Zheng, X. J. (2014). Simulating an AGV scheduling in job workshop for optimal configuration, *Advanced Materials Research*, Vol. 926-930, 1562-1565, doi:[10.4028/www.scientific.net/AMR.926-930.1562](https://doi.org/10.4028/www.scientific.net/AMR.926-930.1562)
- [10] Saidi-Mehrabad, M.; Dehnavi-Arani, S.; Evazabadian, F.; Mahmoodian, V. (2015). An ant colony algorithm (ACA) for solving the new integrated model of job shop scheduling and conflict-free routing of AGVs, *Computers & Industrial Engineering*, Vol. 86, 2-13, doi:[10.1016/j.cie.2015.01.003](https://doi.org/10.1016/j.cie.2015.01.003)
- [11] Aized, T. (2009). Modelling and performance maximization of an integrated automated guided vehicle system using coloured Petri net and response surface methods, *Computers & Industrial Engineering*, Vol. 57, No. 3, 822-831, doi:[10.1016/j.cie.2009.02.009](https://doi.org/10.1016/j.cie.2009.02.009)
- [12] Wang, H.-F.; Chan, C.-H. (2014). Multi-objective optimisation of automated guided dispatching and vehicle routing system, *International Journal of Modelling in Operations Management*, Vol. 4, No. 1/2, 35-62, doi:[10.1504/IJMOM.2014.063585](https://doi.org/10.1504/IJMOM.2014.063585)
- [13] Kawakami, T.; Takata, S. (2012). Battery life cycle management for automatic guided vehicle systems, Matsumoto, M.; Umeda, Y.; Masui, K.; Fukushige, S. (Eds.), *Design for innovative value towards a sustainable society*, Springer, Dordrecht, 403-408
- [14] Oliveira, M. M.; Galdames, J. P. M.; Vivaldini, K.; Magalhaes, D. V.; Becker, M. (2011). Battery state estimation for applications in intelligent warehouses, *IEEE International Conference on Robotics and Automation (ICRA)*, 5511-5516
- [15] Mehta, M. H. (2012). Hybrid genetic algorithm with PSO effect for combinatorial optimization problems, *International Journal of Advanced Computer Research*, Vol. 2, No. 4, 300-305
- [16] Wu, C.-H.; Dong, N.; Ip, W.-H.; Chan, C.-Y.; Yung, K.-L.; Chen, Z.-Q. (2010). Chaotic hybrid algorithm and its application in circle detection, Di Chio, C.; Cagnoni, S.; Cotta, C. et al. (Eds.), *Applications of evolutionary computation*, Springer, Berlin, 302-311
- [17] Mousavi, M.; Yap, H. J.; Musa, S. N.; Tahriri, F.; Dawal, S. Z. M. (2017). Multi-objective AGV scheduling in an FMS using a hybrid of genetic algorithm and particle swarm optimization, *PLoS ONE*, Vol. 12, No. 3, Paper e0169817, doi:[10.1371/journal.pone.0169817](https://doi.org/10.1371/journal.pone.0169817)

- [18] Lau, H. C. W.; Nakandala, D.; Zhao, L. (2015). Development of a hybrid fuzzy genetic algorithm model for solving transportation scheduling problem, *Journal of Information Systems and Technology Management*, Vol. 12, No. 3, 505-524, doi:[10.4301/S1807-17752015000300001](https://doi.org/10.4301/S1807-17752015000300001)
- [19] Taleizadeh, A. A.; Niaki, S. T. A.; Aryanezhad, M.-B.; Shafii, N. (2013). A hybrid method of fuzzy simulation and genetic algorithm to optimize constrained inventory control systems with stochastic replenishments and fuzzy demand, *Information Sciences*, Vol. 220, 425-441, doi:[10.1016/j.ins.2012.07.027](https://doi.org/10.1016/j.ins.2012.07.027)
- [20] Im, S.-M.; Lee, J.-J. (2008). Adaptive crossover, mutation and selection using fuzzy system for genetic algorithms, *Artificial Life and Robotics*, Vol. 13, No. 1, 129-133, doi:[10.1007/s10015-008-0545-1](https://doi.org/10.1007/s10015-008-0545-1)
- [21] Mak, K. L.; Wong, Y. S.; Wang, X. X. (2000). An adaptive genetic algorithm for manufacturing cell formation, *The International Journal of Advanced Manufacturing Technology*, Vol. 16, No. 7, 491-497, doi:[10.1007/s001700070057](https://doi.org/10.1007/s001700070057)
- [22] Egemin Automation Inc. Battery charging systems for automated guided vehicles, from [http://www.egeminusa.com/pages/agvs/agvs\\_battery\\_charging.html](http://www.egeminusa.com/pages/agvs/agvs_battery_charging.html), accessed on 03-03-2016
- [23] AGV KENNIS INSTITUUT. General technology description of AGV-systems, from [http://www.frog.nl/Oplossingen/AGV\\_Kennis\\_Instituut](http://www.frog.nl/Oplossingen/AGV_Kennis_Instituut), accessed on 11-03-2016
- [24] Ghane-Kanafi, A.; Khorram, E. (2015). A new scalarization method for finding the efficient frontier in non-convex multi-objective problems, *Applied Mathematical Modelling*, Vol. 39, No. 23-24, 7483-7498, doi:[10.1016/j.apm.2015.03.022](https://doi.org/10.1016/j.apm.2015.03.022)
- [25] Giagkiozis, I.; Fleming, P. J. (2015). Methods for multi-objective optimization: An analysis, *Information Sciences*, Vol. 293, 338-350, doi:[10.1016/j.ins.2014.08.071](https://doi.org/10.1016/j.ins.2014.08.071)
- [26] Xia, W.; Wu, Z. (2006). A hybrid particle swarm optimization approach for the job-shop scheduling problem, *The International Journal of Advanced Manufacturing Technology*, Vol. 29, No. 3-4, 360-366, doi:[10.1007/s00170-005-2513-4](https://doi.org/10.1007/s00170-005-2513-4)
- [27] Chelouah, R.; Siarry, P. (2003). Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multim minima functions, *European Journal of Operational Research*, Vol. 148, No. 2, 335-348, doi:[10.1016/S0377-2217\(02\)00401-0](https://doi.org/10.1016/S0377-2217(02)00401-0)
- [28] Fan, S.-K. S.; Zahara, E. (2007). A hybrid simplex search and particle swarm optimization for unconstrained optimization, *European Journal of Operational Research*, Vol. 181, No. 2, 527-548, doi:[10.1016/j.ejor.2006.06.034](https://doi.org/10.1016/j.ejor.2006.06.034)
- [29] Kao, Y.-T.; Zahara, E. (2008). A hybrid genetic algorithm and particle swarm optimization for multimodal functions, *Applied Soft Computing*, Vol. 8, No. 2, 849-857, doi:[10.1016/j.asoc.2007.07.002](https://doi.org/10.1016/j.asoc.2007.07.002)
- [30] Liou, C.-D.; Hsieh, Y.-C.; Chen, Y.-Y. (2013). A new encoding scheme-based hybrid algorithm for minimising two-machine flow-shop group scheduling problem, *International Journal of Systems Science*, Vol. 44, No. 1, 77-93, doi:[10.1080/00207721.2011.581396](https://doi.org/10.1080/00207721.2011.581396)
- [31] Tasgetiren, M. F.; Sevkli, M.; Liang, Y.-C.; Gencyilmaz, G. (2004). Particle swarm optimization algorithm for single machine total weighted tardiness problem, *Congress on Evolutionary Computation (CEC 2004)*, 1412-1419
- [32] Kuo, I.-H.; Horng, S.-J.; Kao, T.-W.; Lin, T.-L.; Lee, C.-L.; Terano, T.; Pan, Y. (2009). An efficient flow-shop scheduling algorithm based on a hybrid particle swarm optimization model, *Expert Systems with Applications*, Vol. 36, No. 3, 7027-7032, doi:[10.1016/j.eswa.2008.08.054](https://doi.org/10.1016/j.eswa.2008.08.054)
- [33] Xia, W.; Wu, Z. (2005). An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems, *Computers & Industrial Engineering*, Vol. 48, No. 2, 409-425, doi:[10.1016/j.cie.2005.01.018](https://doi.org/10.1016/j.cie.2005.01.018)
- [34] Song, Y.; Wang, G.; Wang, P.; Johns, A. (1997). Environmental/economic dispatch using fuzzy logic controlled genetic algorithms, *IEEE Proceedings on Generation, Transmission and Distribution*, IET, Vol. 144, 377-382
- [35] Yun, Y.; Gen, M. (2003). Performance analysis of adaptive genetic algorithms with fuzzy logic and heuristics, *Fuzzy Optimization and Decision Making*, Vol. 2, No. 2, 161-175, doi:[10.1023/A:1023499201829](https://doi.org/10.1023/A:1023499201829)