

## **A Genetic Algorithm for Scheduling n Jobs on a Single Machine with a Stochastic Controllable Processing, Tooling Cost and Earliness-Tardiness Penalties**

Mohamed Ali Abdel-Fattah Mansour  
Department of Industrial Engineering, Faculty of Engineering,  
Zagazig University, Zagazig, Sharkia 44519, Egypt

---

**Abstract: Problem statement:** In this research, we addressed the problem of minimizing the earliness-tardiness penalties and manufacturing costs of a single machine with a stochastic controllable processing and tooling cost. **Approach:** We developed a mathematical non-linear integer programming model and its linearised version to find the optimal solution. We introduced a new genome representation in single machine scheduling literature that evolved by a genetic algorithm to solve the problem. The genome representation includes two genes per job, one represents the job starting time and other corresponds to the job processing time. The algorithms were compared based on the solution quality, CPU time and memory consumption in bytes on a set of randomly generated test problems. **Results:** The results showed that developed algorithms could define the global optimal solution of most scheduling problems with  $n \leq 20$  jobs. For larger  $n$ , the developed genetic algorithm outperforms the math models in terms of solution quality and less CPU seconds while consumes moderate memory kilobytes of 3295 compared with 5058 and 1685 of linear and nonlinear models on the average. **Conclusion:** The GA's average performance achieves 6.013 related to the lower bound of math linear program whereas nonlinear model achieves an average of 1.034. The GA's performance increases by increasing  $n$  compared with other techniques. We hope to expand the developed algorithms for different configurations as parallel and job shops.

**Key words:** Single machine scheduling, controllable processing times, earliness-tardiness, mathematical programming, mixed integer, Non-Linear Integer Math Programming (NLIP), Tightness Factor (TF), Genetic Algorithm (GA), linearised model, CPU seconds

---

### **INTRODUCTION**

In this article, we examine the single machine scheduling problem with controllable processing time. In various real-life systems, the job processing time may be controlled by allocating extra resources such as money, manpower, energy, catalysts, spindle speed, feed rate, overtime, subcontracting and so on. For more interesting applications of such scheduling problems, Trick (1994); Kaspi and Shabtay (2003); Wang and Cheng (2005); Kayan and Akturk (2005) and Gurel and Akturk (2007a) for CNC turning operation in flexible manufacturing systems. In relation to these systems, resource allocation and scheduling objectives should be optimized simultaneously to achieve the most efficient system performance.

The modern continuous improvement paradigms such as lean manufacturing and 6-sigma focus on the creation of value through the relentless elimination of waste by the precise allocation of their resources and

producing jobs according to the just-in-time principle. Moreover, these paradigms working on the optimal resource allocations in order to minimize the manufacturing cost that can be reflected in an increase in profit by cost reduction rather than the conventional approach of increasing the profit by price increase. Also, a significant value added to the product price is the inventory holding cost resulting from the excess production that could be eliminated by considering the earliness and lateness penalties. Various research work has been done that combine resource allocations and scheduling objectives as the work of Wassenhove and Baker (1982); Janiak (1987); Daniels and Sarin (1989); Daniels (1990); Panwalkar and Rajagoplan (1992); Cheng *et al.* (1996a; 1996b) and Janiak and Kovalyov (1996).

Vickson (1980) initiated the research in the controllable processing for a single machine with a linear resource consumption function to minimize the total weighted flow time cost plus controllable job

processing cost followed by the work done by Wassenhove and Baker (1982); Janiak (1987) and Janiak and Kovalyov (1996). For a survey concerning this area before 1990 Nowicki and Zdrzalka (1990) and brief surveys given later by Chen *et al.* (1998); Hoogeveen (2005) and Shabtay and Steiner (2007). Hoogeveen and Woeginger (2002) considered a controllable processing single machine scheduling problem to minimize the multi-criteria of the total weighted job processing times plus the linear compression cost function of the processing times and showed that the problem was NP-complete.

Kaspi and Shabtay (2004) considered the single machine scheduling problem with controllable processing time for identical and non-identical job release times restricted by a common limited convex decreasing resource consumption function for minimizing makespan. Moreover, Shabtay and Kaspi (2004) considered the same problem to minimize the total weighted flow time. Janiak *et al.* (2005a) minimized the multi-criteria of the total weighted completion and compression times for a controllable processing single machine. They showed that the problem was equivalent to the half-product minimization problem. Cheng *et al.* (2001) and Ng *et al.* (2005) considered single machine partitioned-jobs group scheduling problems with controllable processing times and the machine processed jobs of the same group simultaneously. Janiak *et al.* (2005b) presented the polynomial time algorithms based on solving two variables linear programming problems by geometric techniques to minimize the total weighted resource consumption of sequencing a set of jobs' groups with independent setup times between groups of a single machine. The setups and processing times are compressible depending on the availability of two resources.

The single machine scheduling problems with resource dependent release times have been extensively studied by Janiak (1985); Cheng and Janiak (1994); Janiak (1998); Janiak (1991) and Li (1994) to minimize a single objective as makespan subjective to the total resource consumption or to minimize the multi-criteria of the total resource consumption cost plus the makespan. The single machine scheduling problem in which both release times and processing times could be controlled by the amount of the resource consumed was recently addressed by Wang and Cheng (2005) to minimize the makespan plus total resource consumption cost.

Although there has been a significant body of research work on process planning decisions for a turning operation with multi-objectives on parallel machine configuration with controllable processing time such as minimizing the sum of makespan and total processing cost as the work done by Alidaee and Ahmadian (1993); Cheng *et al.* (1996a; 1996b); Jansen and Mastrolilli (2004) and Shabtay and Kaspi (2006) there is a little research work done for the single machine scheduling case. Recently, Gurel and Akturk (2007b) considered the minimization of the total manufacturing cost and total completion time simultaneously on identical parallel CNC turning machines. They developed an effective math formulation for the problem by minimizing the total manufacturing cost subject to a given makespan value. Also, they proved some optimality attributes which facilitate efficient heuristic algorithms to generate approximate non-dominated solutions.

Most of the scheduling literature focused on the optimization of scheduling objectives as makespan, total weighted completion times, or solving the multi-criteria objective composed of these two scheduling objectives plus the optimization of the linear resource consumption function. Also, they assumed a discrete compression cost function or linear processing. In a different manner this article focuses on the minimization of earliness and tardiness penalties as scheduling objective plus the manufacturing cost function expressed as a nonlinear convex function of its processing time as showed by Kayan and Akturk (2005). In this article the manufacturing cost is defined in terms of tooling and operating costs.

The purpose of this article is to study the effectiveness of Applying Genetic Algorithms (GAs) to minimize the total manufacturing cost and the earliness-tardiness penalties for manufacturing a set of n jobs of controllable processing times on a single machine. The main reason of selecting GAs is the non-linear nature of the problem that makes it is difficult to find optimal solutions by math programming algorithms when they are always trapped in local optimal solutions. Moreover, GAs are capable of obtaining near to optimal solutions of optimization problems consuming less CPU time and memory bytes (Mansour and Dessouky, 2010). Also, they have the advantage of their flexibility of modeling complex constraints and objective functions. The manufacturing costs include the cost of operating the machine plus the tooling costs. We develop a genome representation that can be considered as a new addition to the single machine scheduling problem with controllable processing and investigate its reliability and applicability for solving large problems.

The remainder of this article is organized as follows. In section 2, we give the problem definition and provide non-linear and linearised math formulation for the problem. The developed genetic algorithm is developed in Section 3. Section 4 provides numerical results on a set of generated test problems compared with the commercial math programming solver. Finally, Section 5 provides the conclusions and future research.

**Problem definition:** The single machine scheduling problem with controllable processing time can be formulated as follows. Let  $\pi = \{1,2,3,\dots,n\}$  be a set of independent and non-preemptive jobs which have to be executed on a single machine and ready to be processed at the start of production period. Each job has a single operation to be performed on the machine and has a uniform random processing time with an upper ( $p_j^u$ ) and lower bounds ( $p_j^l$ ). Also, each job has its due date ( $d_j$ ) and earliness-tardiness penalties ( $a_j$ ) and ( $\beta_j$ ). The cost parameters of performing job  $j$  on the machine were given in terms of machining cost per unit time ( $c_j$ ) of job  $j$  and the tooling cost multiplier ( $m_j$ ) and exponent ( $e_j$ ). The problem consists on allocation of jobs to time slots on the machine and defining the processing time for each one to minimize the sum of earliness-tardiness penalties plus the manufacturing costs.

We use Kayan and Akturk (2005) formula for the manufacturing cost function of producing job  $j$  as a function of  $p_j$  as shown in Eq. 1 below:

$$f_j(p_j) = c_o p_j + m_j p_j^{e_j} \quad (1)$$

The first term is a linear increasing function of  $p_j$  representing the cost of operating the machine for  $p_j$  unit times and the second term is a nonlinear decreasing function of  $p_j$  representing the tooling cost. The parameters  $m_j$  and  $e_j$  representing the job tooling cost multiplier and exponent of job  $j$  whereas the conditions  $m_j > 0$  and  $e_j < 0$  always hold and guarantee that  $f_j(p_j)$  is a non-linear convex function. Moreover, the processing time for job  $j$  is constrained by a lower ( $p_j^l$ ) and upper ( $p_j^u$ ) bounds. Kayan and Akturk (2005) for a detailed description on how  $f_j(p_j)$  is formed and how  $p_j^l$  and  $p_j^u$  are determined.

## MATERIALS AND METHODS

We developed a mixed integer non-linear programming model and its linearised version for the scheduling problem. Also, we developed a genetic

formulation via developing a new genome representation that evolved by a genetic algorithm.

**The mixed integer non-linear math programming model:** The problem can be formulated as a mixed Integer Non-Linear Math Programming (NLIP) model, which considered an extension of the standard single machine scheduling problem that use completion time variables  $C_j$  and the binary variables  $y_{jk}$  to model the problem as the work done by Balas (1984); Queyranne and Wang (1991); Queyranne (1993); Queyranne and Schulz (1994); Pinedo (2002) and Khowala *et al.* (2005), as follows:

$$\text{Minimize } \sum_{j=1}^n \beta_j T_j + \alpha_j E_j + c_o p_j + m_j p_j^{e_j} \quad (2)$$

Subject to:

$$C_j - p_j \geq 0 \quad \forall j \in N \quad (3)$$

$$C_j + p_k \leq C_k + M(1 - y_{jk}) \quad \text{for } j, k \in N \text{ and } j < k \quad (4)$$

$$C_k + p_j \leq C_j + M y_{jk} \quad \text{for } j, k \in N \text{ and } j < k \quad (5)$$

$$p_j \geq p_j^l \quad \forall j \in N \quad (6)$$

$$p_j \leq p_j^u \quad \forall j \in N \quad (7)$$

$$C_j - T_j \leq d_j \quad \forall j \in N \quad (8)$$

$$C_j + E_j \geq d_j \quad \forall j \in N \quad (9)$$

$$C_j \geq 0 \quad \forall j \in N \quad (10)$$

$$y_{jk} \in \{0,1\} \quad \text{for } j, k \in N \text{ and } j < k \quad (11)$$

The nonlinear objective of the mathematical model is to minimize the sum of earliness and tardiness penalties, the manufacturing and tooling costs of  $n$  jobs as indicated by Eq. 2. Equation set (3) ensures that the completion time for each job is greater than or equal to its processing time. The disjunctive constraints are defined in constraints set (4) and (5) that define if the job  $j$  is preceded by job  $k$  or not. The big  $M$  is taken to equal to the sum of upper processing times for all jobs in this article. Equation 6 and 7 define the lower,  $p_j^l$  and upper bounds,  $p_j^u$  for each job. The calculations of the tardiness and earliness penalties are given by satisfying Eq. 8 and 9 respectively. Variables domain is restricted by Eq. 10 and 11.

Gene		s <sub>1</sub>	p <sub>1</sub>	s <sub>2</sub>	p <sub>2</sub>	s <sub>3</sub>	p <sub>3</sub>	s <sub>4</sub>	p <sub>4</sub>	s <sub>5</sub>	p <sub>5</sub>
Possible gene values	Min	0.00	0.81	0.00	1.13	0.00	1.24	0.00	0.98	0.00	0.54
	Max	10.47	0.78	10.47	1.65	10.47	1.63	10.47	0.33	10.47	1.38

Fig. 1: The proposed genome representation

**The mixed integer linear programming model:** The solution of the nonlinear model always trapped in a local optima point so it needs to be linearised to an equivalent math Linear Programming (LP) to find the global optimal solution. The NLIP could be linearised by redefining the term  $p_j$  to the summation shown in Eq. 12 where  $u_i$  is a binary variable define the  $p_j$ 's value,  $p_r$  is the precession of the data set that equals to 0.01 in this article and  $k$  equals to the difference between upper and lower processing time divided by  $p_r$ . Equation 12 represents all possible values of the processing time of job  $j$  and Eq. 13 restricts this value to be a unique in terms of the binary variable  $u_i$ . The term  $p_j^{e_i}$  could be linearised by Eq. 14. Equation 15 defines the variable  $u_i$  as a binary:

$$p_j = \sum_{i=1}^k u_i \times (p_j^L + i \times p_r) \quad \forall j \in N \quad (12)$$

$$\sum_{i=1}^k u_i = 1 \quad (13)$$

$$p_j^{e_i} = \sum_{i=1}^k (p_j^L + i \times p_r)^{e_i} \times u_i \quad \forall j \in N \quad (14)$$

$$u_i \in \{0,1\} \quad \forall i \leq k \quad (15)$$

**The developed GA:** GAs have been widely used in optimization literature to solve the non-polynomial time and complex problems as the satellite daily image selection problem, jobshop scheduling, flexible manufacturing systems operational problems and the optimization of single batch processors of chemical plants. The GA begins by generating a number of genomes equal to a predefined population size and performs a number of evolution processes as crossover, selection, replacement, mutation until satisfying a predefined stopping criteria to define a near to optimal solution. We describe the genetic formulation for the single machine scheduling problem with controllable processing times as follows.

**The genome representation:** The genome representation consists of  $n$  pairs of genes. Each pair

defines the start time and the amount of processing required per job. The genes holds real alleles express time values. For example, consider a 5 jobs problem represented by 10 genes as shown in Fig. 1 where  $s_j$  denotes the possible value for starting the corresponding job. The first pair consists of 2 consecutive genes denoting the start and processing times for job 1. The first gene represents the starting time and includes a value ranges from 0 and max ( $\sum_{j=1}^n p_j^{U, \max(d_j)}$ ) in steps of 0.01. The second represents the processing time required to finish job 1 and so on. For example consider a 5 jobs problem as shown in Fig. 1. Suppose that, the lowers for processing times are {0.81,1.13,1.24,0.98,0.54} and the uppers are {1.59,2.78,2.87,1.31,1.92} then the starting times for all jobs equals any digit ranges from 0 to 10.47 and the alleles for each job processing time are any value ranges from  $p_j^L$  for the job plus a value ranges from 0 to the job corresponding processing difference of {0.78,1.65,1.63,0.33,1.38}.

**The initialization scheme and genetic operators:** The initial population was randomly generated for start times genes by generating a discrete uniformly distributed random number between  $[0, \max(\sum_{j=1}^n p_j, \max d_j)]$  for job  $j$ . For initializing processing time genes, we construct a discrete empirical probability distribution for each job's processing time by dividing the range of processing time to 10 equal intervals.

Let the probability of appearance of the first interval numbers equals 1, the probability equals 0.9 for the second interval numbers and so on. The roulette wheel selection method, two point crossover and swap mutation operators were used to generate new offspring's. The experimental section provides the settings that were used for these operators in this article.

**The genome score:** Our genome representation does not guarantee feasible genomes at the initialization and during the evolution process at each GA's step. To penalize the infeasible genomes, we added a cost term to the objective score  $f$ . It consists of a genome feasibility,  $f_{fes}$  and performance,  $f_{per}$  measures. The genome feasibility part measures the conflict between all jobs and the genome performance part measures the genome total cost. The genome score is calculated as the following procedure:

Step 1: Calculate the sum of processing times of all possible job pair's combinations  $sp_{ij}$

Step 2: Calculate the genome total overlap,  $\delta_g$  that can be defined as the sum of all possible overlaps between any two job pair's combinations between 1 and n. The overlap,  $\delta_{ij}$  between jobs (i, j) can be defined as the difference between the sum of  $p_i$  and  $p_j$  minus the actual difference  $d_{ij}$  as in Eq. 16:

$$\delta_{ij} = p_i + p_j - d_{ij} \quad (16)$$

The actual difference,  $d_{ij}$  between two jobs (i, j) could be determined by Eq. 17. The symbols  $F_i$  and  $S_i$  indicate the finish and start times for jobs i and j respectively:

$$d_{ij} = \max(F_i, F_j) - \min(S_i, S_j) \quad (17)$$

The value of  $\delta_g$  is the sum of all  $\delta_{ij}$  calculated using Eq. 16

Step 3: Calculate the genome feasibility measure,  $f_{fes}$  as in Eq. 18:

$$f_{fes} = (M - \delta_g) / M \quad (18)$$

If  $f_{fes}$  value equals to 1, go to step 4 otherwise go to step 5.

Step 4: Calculate the genome's score  $f = f_{fes}/2.0$  and end

Step 5: Calculate the genome's total cost, TC, that is equals to the sum of tardiness, earliness, operation and tooling costs of the feasible genome then calculate the genome performance measure as shown in Eq. 19:

$$f_{per} = (M - TC) / M \quad (19)$$

Step 6: Calculate the genome score using Eq. 20 and end:

$$f = (f_{fes} + f_{per}) / 2 \quad (20)$$

The GA heuristic was coded using the MATLAB software and tested on a Fujitsu Siemens Laptop, Intel (R) Pentium (R) M with 240 MB RAM, 40 GB HDD, 1.6 GHz speed computer system running Windows XP. The results section will investigate the applicability of using the developed algorithms for solving the scheduling problem under considerations.

## RESULTS AND DISCUSSION

**Data set:** For generating the test problems, five levels for n are considered ranges from 10-50 in steps of 10.

$C_o$  and  $m_j$  factors take the values are generated from a discrete uniform distribution from (0.1,4.5]. The  $e_j$  factor is generated from a discrete uniform distribution from (0.1,2.0]. The Tightness Factor (TF) has two levels of 0.2 and 0.6. The earliness and tardiness penalties ( $\alpha_j$  and  $\beta_j$ ) are generated from a continuous uniform distribution from (0, 100]. The values for  $p_j^L$  and  $p_j^U$  are generated from discrete uniform distribution from (0.4, 3.5] and (0.1, 1.7] respectively. We swap  $p_j^L$  and  $p_j^U$  values if  $p_j^L > p_j^U$  to maintain the solution feasibility. The  $d_j$  values are generated from a discrete uniform distribution from  $(Y \times (1 - TF - 0.5 \times RDD), Y \times (1 - TF + 0.5 \times RDD))$ , where  $Y = 0.5 \times (\sum_{j=1}^n p_j^U + \sum_{j=1}^n p_j^L)$  and RDD (range of due dates) equals 0.2 and 0.6. There are 50 instances for each combinations resulting in 3000 problem instances.

**Computations results:** Table 1 shows the performance of the developed math programming models and the developed GA in terms of the solution quality, CPU seconds and memory kilobytes consumed by each algorithm.

Columns 1-4 depict problem number, n, TF and RDD values for each problem. Columns 5-8 shows the performance of the linearised version of the model in terms of values of best objective,  $O_{best}$ , objective bound,  $O_{bound}$ , CT and Mem where an '\*' sign indicates that the optimal solution was identified. Also, columns 9 to 12 listed the values of  $O_{best}$ ,  $O_{bound}$ , CT and Mem for the non-linear model. The  $O_{best}$  and  $O_{bound}$  are expressed as rational measures with respect to the  $O_{bound}$  of the linearised model.

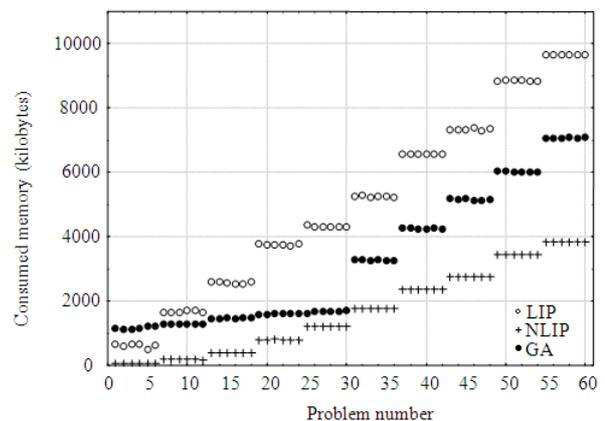


Fig. 2: Scatterplot of problem number and memory consumption

Table 1: The experimental results

Problem characteristics		LINGO LP solution					NLIP solution				GA solution					
n	TF	RDD	O <sub>best</sub>	O <sub>bound</sub>	CT	Mem	O <sub>best</sub>	O <sub>bound</sub>	CT	Mem	O <sub>best</sub>	O <sub>avg</sub>	O <sub>worst</sub>	CT	Mem	(Ps,Ng)
10	0.2	0.2	1.000*	499.648*	52	665	1.054	1.054	4672	62	1.000*	1.105	1.402	24	1136	200,1000
10	0.2	0.6	1.000*	188.141*	45	609	1.001	1.001	170	62	1.000*	1.110	1.414	23	1129	200,1000
10	0.4	0.2	1.000*	194.434*	61	652	1.117	1.117	864	62	1.000*	1.121	1.341	25	1128	200,1000
10	0.4	0.6	1.000*	141.743*	57	654	1.038	1.038	253	62	1.000*	1.131	1.313	26	1138	200,1000
10	0.6	0.2	1.000*	327.038*	312	499	1.028	1.028	7895	62	1.000*	1.141	1.418	25	1201	200,1000
10	0.6	0.6	1.000*	275.250*	55	642	1.005	1.005	731	62	1.000*	1.143	1.423	26	1207	200,1000
20	0.2	0.2	--	283.063	--	1642	4.970	1.065	--	191	1.325	2.435	5.021	521	1272	250,10000
20	0.2	0.6	1.000*	367.293*	2298	1631	1.025	0.669	--	191	1.000*	2.671	5.532	536	1297	250,10000
20	0.4	0.2	2.429	710.984	--	1641	--	0.530	--	191	1.423	3.812	5.731	552	1294	250,10000
20	0.4	0.6	1.000*	276.516*	843	1696	1.112	1.112	15340	189	1.000*	2.641	5.372	512	1275	250,10000
20	0.6	0.2	--	372.834	--	1695	5.710	0.711	--	187	1.643	3.210	5.281	532	1293	250,10000
20	0.6	0.6	1.756	293.529	--	1654	2.523	0.570	--	180	1.438	2.491	5.382	580	1282	250,10000
30	0.2	0.2	--	561.972	--	2585	3.195	0.719	--	408	2.435	3.001	5.985	2219	1444	300,20000
30	0.2	0.6	5.005	415.231	--	2596	5.493	0.870	--	408	2.651	2.998	5.875	2341	1453	300,20000
30	0.4	0.2	22.023	370.823	--	2566	--	0.954	--	408	2.672	2.987	5.829	2254	1483	300,20000
30	0.4	0.6	1.000*	393.328*	15645	2545	--	0.472	--	406	1.000*	3.012	5.972	2301	1449	300,20000
30	0.6	0.2	--	477.303	--	2539	--	0.695	--	401	2.715	3.042	5.871	2306	1473	300,20000
30	0.6	0.6	1.519	514.003	--	2589	--	0.629	--	400	1.072	2.985	5.957	2287	1494	300,20000
40	0.2	0.2	--	546.873	--	3791	--	0.662	--	801	2.987	4.098	8.091	5167	1580	300,30000
40	0.2	0.6	--	266.715	--	3738	--	1.196	--	801	3.009	4.872	7.983	5209	1584	300,30000
40	0.4	0.2	--	312.746	--	3755	--	1.573	--	811	3.098	4.912	8.786	5198	1598	300,30000
40	0.4	0.6	--	285.688	--	3750	13.188	0.960	--	801	2.876	4.761	8.887	5108	1606	300,30000
40	0.6	0.2	--	469.948	--	3705	--	0.841	--	801	2.998	4.981	7.998	5210	1598	300,30000
40	0.6	0.6	--	456.322	--	3791	--	0.610	--	801	2.985	3.998	8.956	5187	1596	300,30000
50	0.2	0.2	--	387.342	--	4362	--	0.917	--	1230	3.987	5.874	8.112	10690	1604	300,40000
50	0.2	0.6	7.880	418.261	--	4302	--	1.003	--	1230	4.076	5.762	8.098	10761	1687	300,40000
50	0.4	0.2	--	264.167	--	4307	--	1.130	--	1230	3.981	4.564	7.982	10782	1685	300,40000
50	0.4	0.6	102.192	267.856	--	4301	--	1.305	--	1233	4.908	6.276	8.002	10921	1669	300,40000
50	0.6	0.2	--	463.310	--	4300	--	0.789	--	1232	3.983	6.256	8.652	10342	1663	300,40000
50	0.6	0.6	--	358.095	--	4298	--	1.019	--	1230	4.009	5.983	8.654	10981	1695	300,40000
60	0.2	0.2	--	208.849	--	5258	--	0.770	--	1758	4.619	6.882	8.349	22903	3296	300,60000
60	0.2	0.6	--	373.296	--	5281	--	1.274	--	1759	4.810	7.038	8.373	22741	3273	300,60000
60	0.4	0.2	--	734.016	--	5224	--	0.868	--	1758	4.353	6.725	8.216	22961	3252	300,60000
60	0.4	0.6	--	224.102	--	5245	--	1.196	--	1758	4.336	5.910	8.286	22987	3271	300,60000
60	0.6	0.2	--	720.396	--	5239	--	0.454	--	1758	5.003	6.613	7.619	22978	3259	300,60000
60	0.6	0.6	--	250.182	--	5229	--	1.479	--	1758	4.711	6.135	8.326	22988	3268	300,60000
70	0.2	0.2	--	433.696	--	6583	--	1.319	--	2378	5.387	7.690	9.724	40744	4282	300,80000
70	0.2	0.6	--	408.991	--	6580	--	1.082	--	2378	5.609	8.450	9.753	41833	4278	300,80000
70	0.4	0.2	--	380.833	--	6573	--	0.736	--	2378	5.078	7.861	9.570	40987	4239	300,80000
70	0.4	0.6	--	284.415	--	6579	--	1.408	--	2369	5.059	6.755	9.651	42810	4229	300,80000
70	0.6	0.2	--	535.399	--	6569	--	0.741	--	2360	5.834	7.625	8.876	40834	4264	300,80000
70	0.6	0.6	--	266.939	--	6559	--	1.367	--	2357	5.495	7.387	9.698	41083	4229	300,80000
80	0.2	0.2	--	552.957	--	7323	--	1.075	--	2749	6.193	8.121	10.296	58957	5192	350,90000
80	0.2	0.6	--	298.201	--	7336	--	1.276	--	2748	6.483	7.740	10.146	59173	5161	350,90000
80	0.4	0.2	--	449.499	--	7339	--	1.233	--	2748	5.879	8.352	10.086	59094	5181	350,90000
80	0.4	0.6	--	218.366	--	7384	--	1.381	--	2747	5.559	7.488	10.170	59018	5131	350,90000
80	0.6	0.2	--	712.547	--	7306	--	1.105	--	2748	6.328	8.103	9.471	58947	5133	350,90000
80	0.6	0.6	--	313.981	--	7342	--	1.388	--	2749	5.890	7.683	10.480	59820	5140	350,90000
90	0.2	0.2	--	363.982	--	8839	--	1.021	--	3439	7.023	9.701	11.809	87155	6039	400,90000
90	0.2	0.6	--	281.718	--	8859	--	1.077	--	3439	7.680	9.317	11.553	87349	6029	400,90000
90	0.4	0.2	--	645.658	--	8855	--	1.219	--	3439	7.379	8.584	10.110	87201	6020	400,90000
90	0.4	0.6	--	378.623	--	8851	--	1.355	--	3439	7.248	9.687	11.723	87903	6019	400,90000
90	0.6	0.2	--	445.356	--	8849	--	1.597	--	3439	7.298	9.015	11.780	88003	6017	400,90000
90	0.6	0.6	--	670.440	--	8819	--	1.131	--	3439	7.077	9.214	11.106	87635	6018	400,90000
100	0.2	0.2	--	530.933	--	9664	--	1.458	--	3846	9.405	11.442	13.646	99543	7064	500,90000
100	0.2	0.6	--	539.364	--	9659	--	1.148	--	3846	10.531	11.863	13.088	100651	7076	500,90000
100	0.4	0.2	--	388.135	--	9660	--	0.984	--	3846	10.038	11.928	13.695	100876	7077	500,90000
100	0.4	0.6	--	184.792	--	9651	--	1.317	--	3846	9.853	10.706	13.221	100899	7080	500,90000
100	0.6	0.2	--	662.755	--	9650	--	1.289	--	3846	9.555	11.617	13.898	101935	7076	500,90000
100	0.6	0.6	--	787.968	--	9645	--	1.034	--	3846	9.427	10.774	12.556	101893	7079	500,90000
	Mean	--	--	--	--	5058	--	1.034	--	1685	4.474	6.013	8.1933	33076	3295	--
	Min.	--	--	--	--	499	--	0.454	--	62	1.000	1.105	1.313	23	1128	--
	Max	--	--	--	--	9664	--	1.597	--	3846	10.531	11.928	13.899	101935	7080	--

The minimum, mean and maximum GA's performances as listed as O<sub>best</sub>, O<sub>avg</sub>, O<sub>worst</sub> in columns 13, 14 and 15 respectively. Columns 16 and 17 show the average computational time and memory consumption for the developed GA. Column 18 lists the Population size (Ps) and number of generation's combinations (Ng) for each problem. The performance of various algorithms are measured by the ratio of the solution/lower bound

found by the algorithm with regard to the lower bound found by the math linear programming model using LINGO software where a stopping criteria of 36,000 seconds was adopted. Each design of Table 1 represents a total of 50 problems, each one solved by the non-linear, linear and the developed GA.

The genetic algorithm parameters such as population size, number of generations, crossover

and mutation probabilities affect the algorithm efficiency. Population sizes and number of generations varied for each test problem as listed in Table 1. The population size was set to 200, 250, 300, 300, 300, 300, 300, 350, 400 and 500 for problems with  $n$  of 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100 respectively. Also, the number of generations was determined based on experimentations for each problem set separately as shown in Table 1 column 18.

Extensive experimental work was done on a hypothetical problem with 40 jobs to find the best values for the crossover and mutation operators. From this test case, a 0.95 crossover and 0.05 mutation probabilities were adopted for all test problems in this article. For each test problem, the GA was run 100 times, each time with a different initial random seed resulting in 300,000 runs for all models.

In general, the linearised version of the nonlinear model could find the global optimal solution of most 10 and 20 jobs designs. Also, the developed GA could find the global optimal solution for these problems consuming lower computational time. Also, it is observed that the NLIP model consumes less memory rather than LP model and GA as shown in Fig. 2. The memory consumed by the non-linear model is so lower that of the linearised model due to the large number of binary variables of the linearised model rather than the non-linear one. Also, the GA consumes larger memory than LP model for the first 6 problems of 10 jobs set and consumes moderate memory kilobytes for all problems.

We first evaluate the developed algorithms for small size problems ( $n = 10$ ), which can be solved optimally by the math LP algorithm using the software package LINGO. Table 1 show that the LP on the average spent 97 seconds to find the optimal solutions while the GA spent less than 25 seconds to find optimal solutions. Also, the NLIP model could not find any optima's and spent on the average 2431 seconds to converge to local optimal solutions. The developed GA could define optimal solutions with a mean average performance of 1.12 compared with the LP bounds.

For medium size problem instances ( $n = 20, 30, 40, 50$ ), the LP and GA heuristics found the optimal solution for problems 8,10 and 16, however the GA spent less time than the LP model to find the optimal solutions and the NLIP algorithm could not find any optimal solutions. Moreover, the LP and the GA could find solutions for 10 problems whereas the NLIP algorithm could find only a local optimal solution for only 8 problems and could not define any optimal solutions for this set of problems. For the other large size problems, none of the math models could define

any candidate best solution but they could define a lower bound for each problem while the developed GA could define feasible and good solutions compared with the lower bounds found by the linear model.

Based on experimentations, we can observe that the GA formulation for the problem under consideration can be solved in less CPU seconds rather than the math formulations of the problem while it consumes moderate memory kilobytes.

## CONCLUSION

In this research, we have proposed a math NLIP model for solving the single machine scheduling problem with controllable processing time for minimizing the total earliness and tardiness penalties and the total manufacturing cost. The developed non-linear model was linearised to obtain the global optimum solution for the scheduling problem. Also, we developed a GA for solving the problem via developing a new genome scheme evolved in a genetic evolution process. The new genome representation could be considered as a new addition to single machine scheduling literature. The developed GA could define the same solutions as the LP model for small size problems where the number of jobs reach to 10 and also outperforms the developed math programming algorithms for larger problems in addition to it consumes small CPU seconds and memory kilobytes.

The natural development in the future of this work is to find a trade-off between the four criteria and to further improve the solution quality and speed of the developed GA. Also, this research can be expanded to different configurations as parallel machines, job shops and flexible job shops.

## REFERENCES

- Alidaee, B. and A. Ahmadian, 1993. Two parallel machine sequencing problems involving controllable job processing times. *Eur. J. Operat. Res.*, 70: 335-341. DOI: 10.1016/0377-2217(93)90245-I
- Balas, E., 1984. *On the Facial Structure of Scheduling Polyhedra*. 1st Edn., Carnegie-Mellon University, Design Research Center, Pittsburgh, pp: 54.
- Cheng, T.C.E. and A. Janiak, 1994. Resource optimal control in some single-machine scheduling problems, *IEEE Trans. Autom. Control*, 39: 1243-1246. DOI: 10.1109/9.293187
- Cheng, T.C.E., Z.L. Chen and C.L. Li, 1996a. Parallel-machine scheduling with controllable processing times, *IIE Trans.*, 28: 177-180. DOI: 10.1080/07408179608966263

- Cheng, T.C.E., Z.L. Chen and C.L. Li, 1996b. Single-machine scheduling with trade-off between number of tardy Jobs and resource allocation, *Operat. Res. Lett.*, 19: 237-242. DOI: 10.1016/S0167-6377(96)00035-1
- Cheng, T.C.E., A. Janiak and M.Y. Kovalyov, 2001. Single machine batch scheduling with resource dependent setup and processing times, *Eur. J. Operat. Res.*, 135: 177-183. DOI: 10.1016/S0377-2217(00)00312-X
- Daniels, R.L. and R.K. Sarin, 1989. Technical note-single machine scheduling with controllable processing times and number of jobs tardy. *Operat. Res.*, 37: 981-984. DOI: 10.1287/OPRE.37.6.981
- Daniels, R.L., 1990. A multi-objective approach to resource allocation in single machine scheduling. *Eur. J. Operat. Res.*, 48: 226-241. DOI: 10.1016/0377-2217(90)90376-M
- Gurel, S. and M.S. Akturk, 2007a. Scheduling parallel CNC machines with time/cost trade-off considerations. *Comput. Operat. Res.*, 34: 2774-2789. DOI: 10.1016/J.COR.2005.10.018
- Gurel, S. and M.S. Akturk, 2007b. Optimal allocation and processing time decisions on non-identical parallel CNC machines:  $\epsilon$ -constraint approach. *Eur. J. Operat. Res.*, 183: 591-607. DOI: 10.1016/J.EJOR.2006.10.024
- Hoogeveen, H. and G.J. Woeginger, 2002. Some comments on sequencing with controllable processing times. *Computing*, 68: 181-192. DOI: 10.1007/s00607-001-1441-x
- Hoogeveen, H., 2005. Multicriteria scheduling. *Eur. J. Operat. Res.*, 167: 592-623. DOI: 10.1016/J.EJOR.2004.07.011
- Janiak, A., 1985. Time-optimal control in a single machine problem with resource constraints. *Automatica*, 22: 745-747. DOI: 10.1016/0005-1098(86)90014-2
- Janiak, A., 1987. One-machine scheduling with allocation of continuously-divisible resource and with no precedence constraints. *Kybernetika*, 23: 289-293.
- Janiak, A., 1991. Single machine scheduling problem with a common deadline and resource dependent release dates. *Eur. J. Operat. Res.*, 53: 317-325. DOI: 10.1016/0377-2217(91)90065-4
- Janiak, A. and M.Y. Kovalyov, 1996. Single machine scheduling subject to deadlines and resource dependent processing times. *Eur. J. Operat. Res.*, 94: 284-291. DOI: 10.1016/0377-2217(96)00129-4
- Janiak, A., 1998. Single machine sequencing with linear models of release dates. *Naval Res. Logistics*, 45: 99-113. DOI: 10.1002/(SICI)1520-6750(199802)45:1<99::AID-NAV6>3.0.CO;2-G
- Janiak, A., M.Y. Kovalyov and M.C. Portmann, 2005a. Single machine group scheduling with resource dependent setup and processing times. *Eur. J. Operat. Res.*, 162: 112-121. DOI: 10.1016/J.EJOR.2002.11.004
- Janiak, A., M.Y. Kovalyov, W. Kubiak and F. Werner, 2005b. Positive half-products and scheduling with controllable processing times. *Eur. J. Operat. Res.*, 165: 416-422. DOI: 10.1016/J.EJOR.2004.04.012
- Jansen, K. and M. Mastrolilli, 2004. Approximation schemes for parallel machine scheduling problems with controllable processing times. *Comput. Operat. Res.*, 31: 1565-1581. DOI: 10.1016/S0305-0548(03)00101-1
- Kaspi, M. and D. Shabtay, 2003. Optimization of the machining economics problem for a multistage transfer machine under failure, opportunistic and integrated replacement strategies. *Int. J. Product. Res.*, 41: 2229-2247. DOI: 10.1080/0020754031000087175
- Kaspi, M. and D. Shabtay, 2004. Convex resource allocation for minimizing the makespan in a single machine with job release dates. *Comput. Operat. Res.*, 31: 1481-1489. DOI: 10.1016/S0305-0548(03)00103-5
- Kayan, R.K. and M.S. Akturk, 2005. A new bounding mechanism for the CNC machine scheduling problems with controllable processing times. *Eur. J. Operat. Res.*, 167: 624-643. DOI: 10.1016/J.EJOR.2004.07.012
- Khowala, K., A. Keha and J. Fowler, 2005. A comparison of different formulations for the non-preemptive single machine total weighted tardiness scheduling problem. Arizona State University, Tempe, USA. <http://www.mistaconference.org>
- Li, C.L., 1994. Scheduling with resource-dependent release dates-a comparison of two different resource consumption functions. *Naval Res. Logistics*, 41: 807-819. DOI: 10.1002/1520-6750(199410)41:6<807::AID-NAV3220410609>3.0.CO;2-8
- Mansour, M.A.A. and M.M. Dessouky, 2010. A genetic algorithm approach for solving the daily photograph selection problem of the SPOT5 satellite. *Comput. Indus. Eng.*, 58: 509-520. DOI: 10.1016/J.CIE.2009.11.012

- Ng, C.T., T.C.E. Cheng, A. Janiak and M.Y. Kovalyov, 2005. Group scheduling with controllable setup and processing times: Minimizing total weighted completion time. *Annals Operat. Res.*, 133: 163-174. DOI: 10.1007/S10479-004-5030-1
- Nowicki, E. and S. Zdrzalka, 1990. A survey of results for sequencing problems with controllable processing times. *Discrete Applied Math.*, 26: 271-287. DOI: 10.1016/0166-218x(90)90105-L
- Panwalkar, S.S. and R. Rajagoplan, 1992. Single-machine sequencing with controllable processing times. *Eur. J. Operat. Res.*, 59: 298-302. DOI: 10.1016/0377-2217(92)90144-X
- Pinedo, M., 2002. *Scheduling: Theory, Algorithms and Systems*. 2nd Edn., Prentice-Hall, Upper Saddle River, NJ., ISBN: 0130281387, pp: 586.
- Queyranne, M. and Y. Wang, 1991. Single-machine scheduling polyhedra with precedence constraints. *Math. Operat. Res.*, 16: 1-20. DOI: 10.1287/MOOR.16.1.1
- Queyranne, M., 1993. Structure of a simple scheduling polyhedron. *Math. Prog.*, 58: 263-285. DOI: 10.1007/BF01581271
- Queyranne, M. and A.S. Schulz, 1994. *Polyhedral approaches to machine scheduling*. CiteSeerx. <http://citeseerx.ist.psu.edu>
- Shabtay, D. and M. Kaspi, 2004. Minimizing the total weighted flow time in a single machine with controllable processing times. *Comput. Operat. Res.*, 31: 2279-2289. DOI: 10.1016/S0305-0548(03)00187-4
- Shabtay, D. and M. Kaspi, 2006. Parallel machine scheduling with a convex resource consumption function. *Eur. J. Operat. Res.*, 173: 92-107. DOI: 10.1016/J.EJOR.2004.12.008
- Shabtay, D. and G. Steiner, 2007. A survey of scheduling with controllable processing times. *Discrete Applied Math.*, 155: 1643-1666. DOI: 10.1016/J.DAM.2007.02.003
- Trick, M.A., 1994. Scheduling multiple variable-speed machines. *Operat. Res.*, 42: 234-248. DOI: 10.1287/OPRE.42.2.234
- Vickson, R.G., 1980. Choosing the job sequence and processing times to minimize total processing plus flow cost on a single machine. *Operat. Res.*, 28: 1155-1167. DOI: 10.1287/OPRE.28.5.1155
- Wassenhove, L.N.V. and K.R. Baker, 1982. A bicriterion approach to time/cost trade-offs in sequencing. *Eur. J. Operat. Res.*, 11: 48-54. DOI: 10.1016/S0377-2217(82)80008-8
- Wang, X. and T.C.E. Cheng, 2005. Single machine scheduling with resource dependent release times and processing times. *Eur. J. Operat. Res.*, 162: 727-739. DOI: 10.1016/J.EJOR.2003.10.024