

## Solving the Periodic Maintenance Scheduling Problem via Genetic Algorithm to Balance Workforce Levels and Maintenance Cost

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

---

**Abstract: Problem statement:** In this article we address the multi-objective Periodic Maintenance Scheduling Problem (PMSP) of scheduling a set of cyclic maintenance operations for a given set of machines through a specified planning period to minimize the total variance of workforce levels measured in man-hours and maintenance costs with equal weights. **Approach:** The article proposed a mixed integer non-linear math programming model and a linearised model for the PMSP. Also, we proposed a Genetic Algorithm (GA) for solving the problem using a new genome representation considered as a new addition to the maintenance scheduling literature. The algorithms were compared on a set of representative test problems. **Results:** The developed GA proves its capability and superiority to find good solutions for the PMSP and outperforms solutions found by the commercial optimization package CPLEX. The results indicated that the developed algorithms were able to identify optimal solutions for small size problems up to 5 machines and 6 planning periods. The GAs defined solutions in 22 seconds consuming less than two kilobytes with a reliability of 0.84 while the nonlinear and linear models consumes on average 705 and 37 kilobytes respectively. **Conclusion:** The developed GA could define solutions of average performance of 0.34 and 0.8 for the linearized algorithm compared with lower bound defined by the nonlinear math programming model. We hope to expand the developed algorithms for integrating maintenance planning and aggregate production planning problems.

**Key words:** Periodic Maintenance, multi-criteria optimization, mixed-integer non-linear math programming, linearization, genetic algorithms, maintenance costs, complexity parameter, CPU seconds, genome's fitness

---

### INTRODUCTION

The preventive maintenance scheduling is among the most important problems faced by productive/service organizations. The preventive maintenance applied by servicing the equipment on regular intervals is for the purpose of increasing its reliability as much as possible. The problem has attracted researchers due to its economical importance and complexity, see for example Dekker (1996). These articles and others contained therein were interested in modeling and solving the problem to minimize the cost or maximize the machine lifetime. Extensive research treated the problem as a stochastic model whereas the machine failures described by probability distributions (see for instance Gertsbakh and Gertsbakh (2000) while a little was concerned with the deterministic case where the failures described by constant parameters (see Wagner *et al.* (1964) for an early reference).

Two variants of maintenance planning problems exist in literature, the Free Periodic Maintenance

Scheduling Problem (FPMSP) and the Periodic Maintenance Scheduling Problem (PMSP). The FPMSP considers  $T$  as a decision variable and aims to find the optimal  $T^*$  optimizing maintenance costs while the PMSP assumes values of cycle lengths. Bar-Noy *et al.* (2002) have shown that the FPMSP is NP-hard even when  $T^*$  is known so this implies the PMSP is NP-hard (Grigoriev *et al.*, 2006).

The PMSP is to find a cyclic maintenance schedule of a given length  $T$  for a set of machines to optimize predefined organization goals. The maintenance scheduling problems research work could be broadly categorized as stochastic and deterministic approaches based on the machine failures pattern. The nature of machine failures in stochastic approaches is described by probability distributions (see for instance Gertsbakh and Gertsbakh (2000)) while the deterministic approaches are used to describe the cyclic maintenance scheduling activities assuming a fixed cycle length. The stochastic machine failure maintenance scheduling has attracted researchers more than the deterministic case.

A precise description for the problem is given in the following.

**Problem description:** We consider the following practical problem. There is a set of  $m$  machines types, each includes  $m_i$  identical ones where  $i \in \{1, 2, \dots, m\}$  and a standard machine was defined based on the experience of maintenance expertise. The standard machine maintenance cycle was known and represents all required maintenance operations during a predefined period. The cycle length ( $T$ ) equals to the total number of maintenance activities for the standard unit. The workforce levels measured by the total man-hours and the required cash flow to perform each maintenance operation are given. A complexity parameter ( $C_{ki}$ ) for machine number  $l$  of type  $k$  given for all machines represents the similarity between each machine with the standard unit. The work content for all machines could be estimated by referencing to the machines complexity parameters. The maintenance laborers are hired all the year so it is fixed. A solution to the problem has information on the cyclic schedule and the allocated cost for each machine through  $T$ .

To illustrate the problem, consider a PSMP with 2 machine types A and B each has 3 machines symbolized as A1, A2, A3, B1, B2 and B3. The standard machine A1 has a cycle of C-I<sub>1</sub>-I<sub>2</sub>-I<sub>3</sub>-S<sub>1</sub>-I<sub>4</sub>-I<sub>5</sub>-I<sub>6</sub>-S<sub>2</sub>-I<sub>7</sub>-I<sub>8</sub>-I<sub>9</sub> ( $T = 12$ ). C indicates an overhaul S<sub>1</sub> indicates a half overhaul, I<sub>1</sub>-I<sub>8</sub> indicate various maintenance operations. Each machine complexity parameter is given in Table 1. 1800, 200 and 100 man-hrs are required to perform C, S<sub>1</sub> and I<sub>1</sub> respectively. B1 machine has a complexity parameter of 0.5, meaning that if any operation performed with 200 man-hours for machine A1 (the standard one), it will be performed on 100 man-hours for B1. C, S<sub>1</sub> and I<sub>1</sub> costs are 1000, 80 and 50 respectively for machine A1. The associated costs for B1 are 500, 40 and 25 respectively. Table 1 depicts a random schedule of man-hours and money with  $T = 12$ . Each machine has the standard machine's maintenance cycle that begins by any maintenance operation C, any of Ss, or any of Is and follow the cycle sequence. The table depicts a fluctuation in monthly man-hours to perform all maintenance work between 950 and 17100. Also, the monthly money allocated to maintenance operations varies from 525-\$10500. This solution has man-hours and cost variances of 19623108 and 7328792 with a total of 26,951,900. A possible heuristic solution approach for the PMSP implemented by the author for many years in the field of maintenance scheduling is to assign an overhaul to the first machine at the first period and an overhaul to the second at period 2 and so on. The variances for man-hours and

costs are 4227969 and 1361584 with a total variance of 5,589,553. For a set of 15 practical cases of 10 machines and  $T = 12$ , we solved each case by both random and heuristic approaches. The resulting variance for each case is depicted at the scatter plot of Fig. 1. The scatter plot shows that the random plan where we begin the cycle for each machine by a random maintenance operation always gives larger variance than the heuristic approach.

Our motivations for investigating the PMSP rather than FPMSP are threefold. First of all, the PMSP is a more practical and industrial problem than the academic FPMSP where the number of machines may reach to 500 or more and the cycle length might be 250, 52, 30, 7, 24 or 60. The second motivation of this study is solving the PMSP to optimality whereas most previous research work was focused on the complexity and approximation of the PMSP. A third motivation of our work is to consider a real objective of smoothing the workforce levels and maintenance costs through  $T$  using a non-linear objective function (variance).

The purpose of this study is to introduce Genetic Algorithms (GAs) as a solver for the PMSP with the objective of minimizing the total variance of workforce man-hours and maintenance costs with equal weights. GAs include features for handling a large number of constraints, modeling flexibility in dealing with problem complexity, utilizing less CPU memory allocation than mathematical programming techniques, handling the multi-criteria nature of real-world problems without any modeling complications and ease of implementation. Furthermore, there have been a significant number of studies that have shown GAs effectiveness in solving hard combinatorial problems. For example, Beasley and Chu (1996) and Lorena and Lopes (1997) applied GA to the set covering problem. Also, we proposed a mixed-integer non-linear math programming model and a linearized model for the PMSP to evaluate the performance of the developed GA. The evaluation was based on the solution defined by math models due to the concerned problem differ from the standard PMSP so the existing solution methods must developed to handle this difference.

In summary, the two primary contributions of this article to the PMSP literature are that: (1) we demonstrate the effectiveness of GAs for solving the PMSP over the developed math programming models; (2) we handle the problem as a non-linear objective optimization problem rather than the available literature of handling only the linear cases and deal with the practical rather than the academic problems available in maintenance planning and scheduling literature.

Table 1: The distribution of man-hours and costs through T = 12

Machine		T											
Type	C <sub>kl</sub>	1	2	3	4	5	6	7	8	9	10	11	12
A1	1	C	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>	S <sub>1</sub>	I <sub>4</sub>	I <sub>5</sub>	I <sub>6</sub>	S <sub>2</sub>	I <sub>7</sub>	I <sub>8</sub>	I <sub>9</sub>
		1800	100	100	100	200	100	100	100	200	100	100	100
A2	1.5	S <sub>2</sub>	I <sub>7</sub>	I <sub>8</sub>	I <sub>9</sub>	C	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>	S <sub>1</sub>	I <sub>4</sub>	I <sub>5</sub>	I <sub>6</sub>
		1000	50	80	70	80	50	70	80	80	80	50	60
A3	2	S <sub>1</sub>	I <sub>4</sub>	I <sub>5</sub>	I <sub>6</sub>	S <sub>2</sub>	I <sub>7</sub>	I <sub>8</sub>	I <sub>9</sub>	C	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>
		300	150	150	150	300	150	150	150	2700	150	150	150
B1	0.5	I <sub>3</sub>	S <sub>1</sub>	I <sub>4</sub>	I <sub>5</sub>	I <sub>6</sub>	S <sub>2</sub>	I <sub>7</sub>	I <sub>8</sub>	I <sub>9</sub>	C	I <sub>1</sub>	I <sub>2</sub>
		120	120	75	90	1500	75	120	105	120	120	75	105
B2	2.5	S <sub>1</sub>	I <sub>4</sub>	I <sub>5</sub>	I <sub>6</sub>	S <sub>2</sub>	I <sub>7</sub>	I <sub>8</sub>	I <sub>9</sub>	C	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>
		400	200	200	200	3600	200	200	200	400	400	200	200
B3	3	I <sub>3</sub>	S <sub>1</sub>	I <sub>4</sub>	I <sub>5</sub>	I <sub>6</sub>	S <sub>2</sub>	I <sub>7</sub>	I <sub>8</sub>	I <sub>9</sub>	C	I <sub>1</sub>	I <sub>2</sub>
		160	100	140	160	160	160	100	120	2000	100	160	140
ΣMan-hours		17100	950	950	950	1900	950	950	950	1900	950	950	950
ΣCost,\$		10500	525	840	735	840	525	735	840	840	525	630	

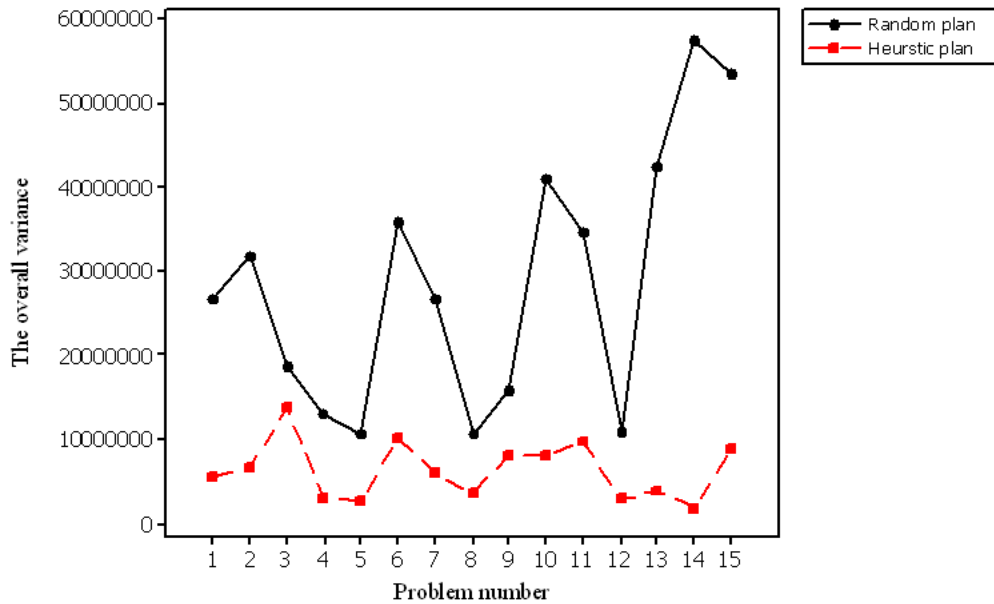


Fig. 1: A scatter plot of variances for 15 practical cases

**Literature review:** The preventive maintenance scheduling models and solution widely see techniques of power generating plants have been considered in scheduling literature, for example, the article by Sherif and Smith (1981), Dekker (1996) and Dekker *et al.* (1997). Presented an overview for the optimization techniques including integer programming models for optimizing preventive maintenance scheduling problems of power generating plants. Also, Charest and Ferland (1993) applied local search techniques to solve

the set-partitioning model for power generating units. Duffuaa and Ben-Daya (1994), Hariga (1994) and Sule and Harmon (1979) investigated preventive maintenance scheduling models for the purpose of coordinating a common resource to a set of machines.

Anily *et al.* (1998) considered a special case of FPMSP for scheduling of maintenance service to a set of machines applied in a multi-item replenishment of stock environment. They assumed linear increase of maintenance cost coefficients. They prove the existence

of a cyclic maintenance schedule for FPMP and prove that the problem has an exact solution by introducing an exponential complexity network-flow based algorithm that could find the exact solution for up to 4 machines problems. Anily *et al.* (1999) proposed an algorithm for scheduling of maintenance services to three machines and found the exact optimal solutions for certain instances of the problem and the other instances provided a heuristic algorithm with a performance ratio of 1.0333.

Bar-Noy *et al.* (2002) and Kenyon *et al.* (2000) constrained the number of machines that can be served at each period to be at most  $M$  in a broadcast scheduling environment. Bar-Noy *et al.* (2002) proved the NP-hardness of the FPMP and investigated lower bounds and proposed a  $\frac{9}{8}$  approximation algorithm.

Also, Schabanel (2000) showed that the preemptive FPMP was also NP-hard. Kenyon *et al.* (2000) introduced a Polynomial-time approximation scheme for data broadcast of a bounded service costs environment while the non-identical service times case was studied by Kenyon and Schabanel (2001). Brauner *et al.* (2001) addressed the complexity of high multiplicity scheduling problems that arise from compact encodings of solutions.

A dynamic programming technique to minimize the salvage costs and the discounting of costs of operating a parallel machine shop whose operational costs increase with age. The problem deals with the possibility of replacing a machine at each planning period at the expense of purchasing costs.

Based on the previous survey, we note that no research work done tackling the PMSP with a large number of machines corresponds to real-world manufacturing systems. Also, no research was done to minimize the total variance workforce man-hrs and maintenance costs. Moreover, previous research focused on the parallel machine problem when at each planning period there was one machine in maintenance and other machines in service but this study allowed for more than one machine to be in maintenance.

## MATERIALS AND METHODS

**The solution methodology:** In this article, we adopt a solution methodology shown in Fig. 2. The methodology consists of four steps. The first step begins by generating a set of test problems by considering the factors affecting the complexity of the PMSP and their levels. The system size, cycle length and the number of different maintenance operations were considered. A full experimental design was

adopted to generate the full factorial design for the factors. Each design was replicated 50 times to generate 50 instances per design.

A standard time-indexed mixed-integer non-linear math programming model was developed in the second step to represent the PMSP for minimizing the total variance of man-hours and costs through the planning period with equal weights. The model considers the maintenance plan sequence of operations for all machines, the total of man-hours per period must not exceed the available and the total maintenance cost doesn't exceed the available money allocated to perform maintenance operations. The software package LINGO was applied to solve all non-linear models. A linear integer math programming model version for the original PSMP was developed by linearizing the nonlinear terms in the objective function and adding the necessary constraints to find the global optimal solution. The software CPLEX was used to find the linear solutions. Applying integer math programming solvers to handle practical problems usually has many practical implementation difficulties such as higher CPU seconds, memory bytes and needs more modeling effort to code the problem in the math form.

GAs were used extensively in scheduling literature to solve these difficulties whereas they do not need high amount of CPU seconds, consume low memory bytes compared with math programming solvers and they do not need a lot of coding effort to formulate the problem in genetic form. Moreover, the math programming solvers for large-size problems might fail to find any feasible solution in suitable CPU seconds, whereas the GAs find a feasible solution during the initial evolution process. Unfortunately, GAs need their parameter tuning to find better solutions for complex problem structures.

The third step is concerned with designing the developed GA by conducting a full factorial design for the factors affecting the GA's efficiency. Each factor levels were defined and the GA was run for 100 replicates per design. The parameter's selection criterion was the best average performance for minimizing the variance. After the GA's parameter tuning step, we applied the developed GA in the last step to find the solution of the generated test problems at step 1. Each problem was solved 100 times to find the minimum, average and maximum variances, reliability to find the minimum variance, average consumed CPU seconds and the average memory bytes were traced with respect to different population sizes and number of generation combinations.

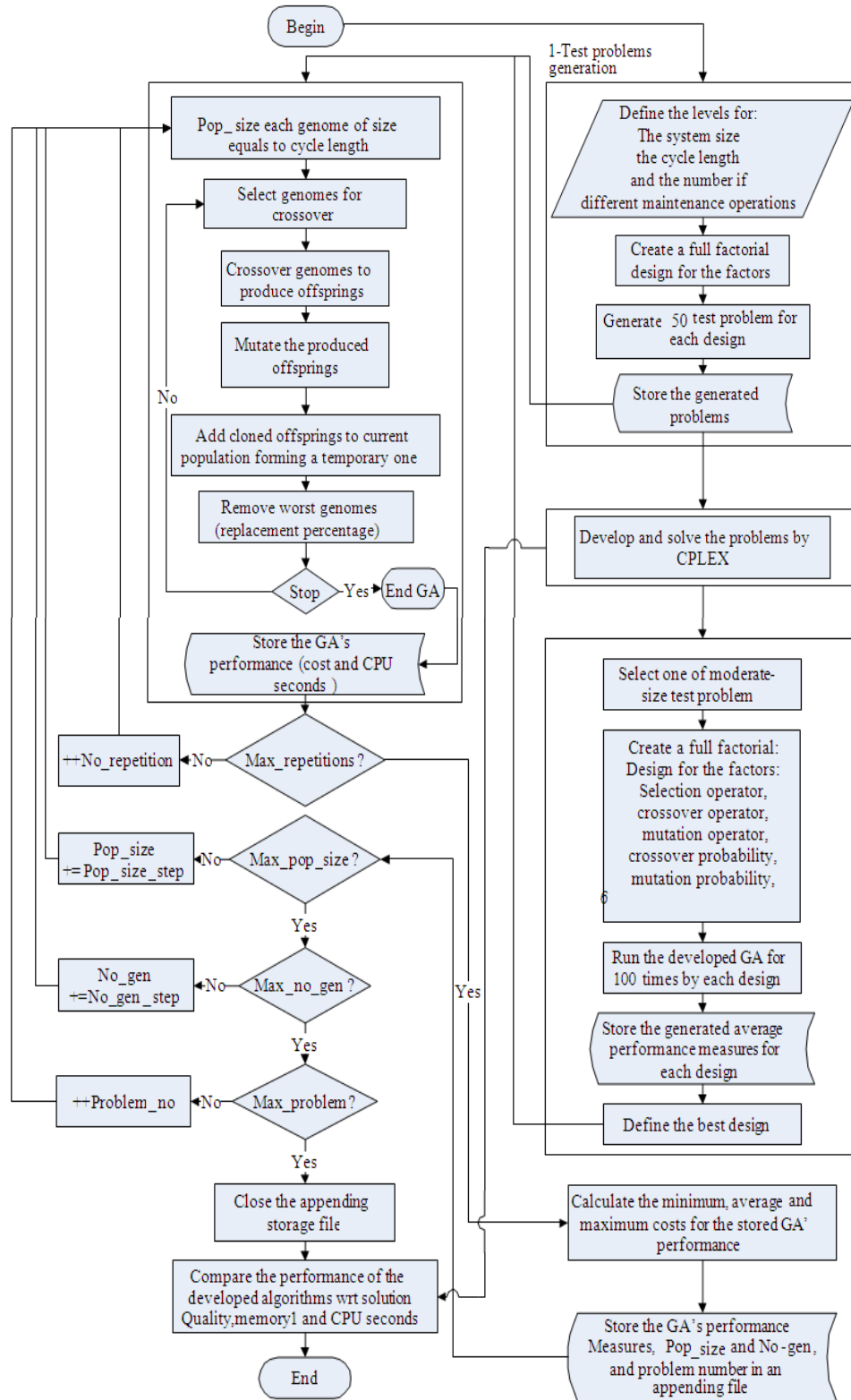


Fig. 2: The solution methodology

By comparing the three developed solution models for the PSMP, we could find the limitations, advantages, disadvantages and applicability of using each model to solve real-world maintenance scheduling problems.

We will propose a mixed-integer non-linear math programming model for the PSMP and the linearised version of the model.

**The math programming models:** We develop a mixed-integer non-linear math programming model for PMSP as a standard time-indexed formulation and linearise it to find the global optimal solution. The notations used in this formulation are listed below.

**Indexes:**

- i = Index for the maintenance operation, I = 1,2,3,...T
- j = Index for the planning period, J = 1,2,3,...,T
- k = Index for the machine type, K = 1,2,3,...,m
- l = Index for the number of machine per type k,l=1,2,3,...,m<sup>k</sup>

**Parameters:**

- a = Weight to balance man-hours and maintenance cost variances and 0 ≤ a ≤ 1. It equals to 0.5 in this article
- c<sub>j</sub> = The cost for maintenance operation j of the standard machine
- C<sub>j</sub> = The available money at period j for maintenance operations
- c<sub>kl</sub> = Complexity parameter for machine number l of type k
- h<sub>ikl</sub> = The required man-hours to perform maintenance operation i for machine number l of type k and equals to h<sub>j</sub>, c<sub>kl</sub>
- h<sub>j</sub> = The required man-hours to perform operation j for the standard machine
- H<sub>j</sub> = The available man-hours at period j
- M = Number of machine types
- m<sub>k</sub> = Number of machines per type k
- T = Cycle length

**Decision variables:**

- x<sub>ijkl</sub> = A binary decision variable equals to 1 if a maintenance operation i will be performed at period j of the machine type k for the machine number l and equals to 0 otherwise
- y<sub>ikl</sub> = An integer variable represents the completion time of maintenance operation i for the machine number l of type k

z<sub>ijkl</sub> = A binary decision variable equals to 1 if any maintenance operation will be performed at period j or the machine number l of type k and equals to 0 otherwise

Based on the definitions of indexes, parameters and decision variables we formulate the multi-objective mixed-integer non-linear math programming model for the PSMP as follows:

$$\text{Min } \alpha \cdot \left( \frac{1}{T} \cdot \sum_{j=1}^T \left( \sum_{k=1}^m \sum_{l=1}^{m_k} \sum_{i=1}^T c_{kl} \cdot h_j \cdot x_{ijkl} - \left( \frac{1}{T} \cdot \sum_{j=1}^T \sum_{k=1}^m \sum_{l=1}^{m_k} \sum_{i=1}^T c_{kl} \cdot h_j \cdot x_{ijkl} \right) \right) \right)^2 + (1-\alpha) \cdot \left( \frac{1}{T} \cdot \sum_{j=1}^T \left( \sum_{k=1}^m \sum_{l=1}^{m_k} \sum_{i=1}^T c_{kl} \cdot c_j \cdot x_{ijkl} - \left( \frac{1}{T} \cdot \sum_{j=1}^T \sum_{k=1}^m \sum_{l=1}^{m_k} \sum_{i=1}^T c_{kl} \cdot c_j \cdot x_{ijkl} \right) \right) \right)^2 \tag{1}$$

Subject to:

$$\sum_{k=1}^m \sum_{l=1}^{m_k} \sum_{i=1}^T c_{kl} \cdot h_j \cdot x_{ijkl} - H_j \leq 0 \quad \forall j = 1, 2, \dots, T \tag{2}$$

$$\sum_{k=1}^m \sum_{l=1}^{m_k} \sum_{i=1}^T c_{kl} \cdot c_j \cdot x_{ijkl} - C_j \leq 0 \quad \forall j = 1, 2, \dots, T \tag{3}$$

$$\sum_{j=1}^T j \cdot x_{ijkl} - y_{ikl} = 0 \quad \forall i = 1, 2, \dots, T; \forall k = 1, 2, \dots, m; \forall l = 1, 2, \dots, m_k \tag{4}$$

$$\sum_{j=1}^T x_{ijkl} - T \cdot z_{ijkl} = 0 \quad \forall i = 1, 2, \dots, T; \forall k = 1, 2, \dots, m; \forall l = 1, 2, \dots, m_k \tag{5}$$

$$\sum_{i=1}^T z_{ikl} = 1 \quad \forall k = 1, 2, \dots, m; \forall l = 1, 2, \dots, m_k \tag{6}$$

$$x_{ijkl}, z_{ikl} \in \{0,1\}$$

$$y_{ikl} \geq 0 \text{ and integers:}$$

$$\forall i = 1, 2, \dots, T; j = 1, 2, \dots, T; \forall k = 1, 2, \dots, m; \quad (7)$$

$$\forall l = 1, 2, \dots, m_k$$

The objective function shown in Eq. 1 minimizes the total variances of man-hours and costs for planning periods T. Eq. 2 ensures that the total allocation of man-hours per period doesn't exceed the available per period J. Equation 3 ensures that the total allocated costs per period are not more than the available. Equations 4-6 ensure that maintenance operations assignment follow the sequence of operations for the standard unit maintenance cycle. The variable domains are defined by Eq. 7.

In this article, we reformulate the model to a mixed-integer linear formulation by performing the following variable transformations. For each product of binary variable pairs  $x_{ijkl} \cdot x_{pqrs}$  in the objective function, replace  $x_{ijkl} \cdot x_{pqrs}$  by  $z_{ijklpqrs}$  and adding the following three constraints as shown in Eq. 8-11:

$$x_{ijkl} + x_{pqrs} - 2 \cdot z_{ijklpqrs} \geq 0 \quad (8)$$

$$x_{ijkl} + x_{pqrs} - z_{ijklpqrs} \leq 1 \quad (9)$$

$$z_{ijklpqrs} \in \{0,1\} \quad (10)$$

The linearised model can find the global optimal solution for the non-linear model for small-size problems and also can find lower bounds for larger problems. We will introduce a GA formulation for the PSMP by originating a new genome representation for minimizing the total variance of man-hours and costs.

**The proposed genetic algorithm:** The GA is an optimization heuristic based on a stochastic search which mimics the biological process of natural selection (Goldberg, 1989; Davis, 1991). It has widely applied to solve hard single and multi-criteria optimization problems as job-shop scheduling, flexible manufacturing loading, transportation planning, daily image selection problem (Mansour and Dessouky, 2010). The algorithm evolves with a constant population size for a predefined period defined by a stopping criterion. This criterion may be a fixed number of generations, a certain execution time, or until an occurrence of a population convergence. The GA begins by generating an initial population of individuals

known as genomes. Each genome represents a feasible solution for the problem and has a fitness controlling its existence in the next generations. At each generation, a new population is formed by mating genomes as a result of applying genetic operators as crossover and mutation so the good genomes replace the worst ones in the current generation (Michalewicz, 1996). A systematic GA's structure is depicted in Fig. 2.

We will develop a GA to provide efficient solutions to the predefined PSMP. We first begin with introducing a new genome representation for the PSMP and then a detailed GA's structure will be defined in terms of initialization, selection, crossover and mutation and replacement mechanisms. Also, the genome's fitness evaluation is illustrated besides conducting an experiment to define the most affecting GA's components on maximizing the algorithm performance.

**Genome representation, initialization, selection and genetic operators:**

The proposed genome consists of number of genes equal to the total number of machines in the manufacturing system,  $\sum m_k$  for all values of k. Each gene's allele was generated from a discrete uniform distribution with lower and upper bounds equals to 1 and T respectively. The gene order in the genome sequence represents the machine number while the gene's allele represents the maintenance operation that will be performed on that machine at period 1 of the planning horizon. For example, the allele value at position 2 of a hypothetical genome represents the first maintenance operation that will be performed at period 1 for machine number 2. Figure 3 illustrates an industrial system consisting of 5 machines and it is required to construct a maintenance schedule for 6 planning periods. The first machine (M1) chosen as a standard machine and has maintenance schedule of  $O_1, O_2, \dots, O_6$ , while the third machine schedule is  $O_2, O_3, O_4, O_5, O_6, O_1$  and so on. The depicted schedule could be presented as the genome  $O_1, O_6, O_2, O_5, O_4$ . The genome indicates that maintenance operations 1, 6, 2, 5 and 4 will be adopted for machines 1 to 5 as the starting maintenance operation at period 1. The proposed genome advantages are the small length, avoiding the representation's infeasibility that exhausts GAs and existence of the standard genetic operators.

The initial population was randomly generated and controlled by various genetic operators through the evolution process. Different selection operators were adopted as tournament, ranking, roulette wheel and uniform. One point, two point and uniform crossover operators were applied. The flip and Gaussian mutation operators were applied in the developed GA.

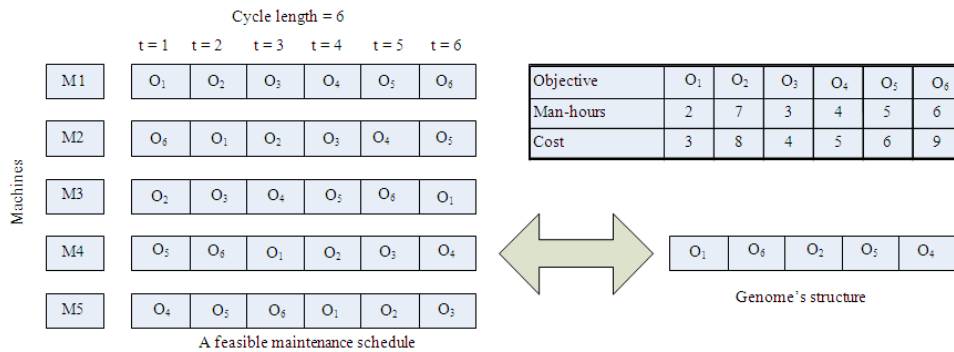


Fig. 3: The proposed genome representation

**Genome's fitness evaluation:** The genome's fitness determines the power to achieve the objectives of solving the problem. It controls the genome's existence through the evolution process. In this article, the fitness evaluation process evaluates the total variance of man-hours loads and costs per planning period. The variance was computed by the formula shown in Eq. 11. The  $X_j$  variable represents the total man-hours or costs allocated by a plan at period  $j$  while  $\bar{x}$  is the overall average of all periods and is the average per period  $j$ :

$$\text{Variance} = \frac{\sum_{j=1}^T (\bar{x}_j - \bar{x})^2}{T} \tag{11}$$

For the shown example in Fig. 3, the total man-hours for periods are 24, 23, 22, 21, 25 and 20 respectively. The overall average of 22.5 man-hours is achieved for the given schedule and variance equals to 0.324. Also, the cost's variance equals to 0.179. The total of these two variance components gives the genome's objective.

## RESULTS AND DISCUSSION

**Genetic algorithm parameter settings:** To identify the best GA's parameters for the PMSP, we conducted a full experimental design on 8 factors. The eight factors were tested on 5 randomly generated test problems including 10,20,30,40 and 50 machines and a planning period of 52 weeks. The experiment's objective is to define the most effective level for each factor and the most effective factor combinations associated with the difference in the GA's behavior. The factors are selection, crossover, mutation schemes, crossover and mutation probabilities, replacement percentage, population size and number of generations. It is indicated that these factors affects the algorithm efficiency. The population size and number of

generations varied for each problem hence they will be defined for each one separately. Based on the experiments the factors settings of uniform crossover probability of 0.93, flip mutation probability of 0.01 and roulette wheel replacement percentage of 0.25, give the best average total variance. For each test problem, the genetic algorithm was run 100 times, each time with a different initial random seed.

We will test the developed math programming and GA models on a set of test problems considering the most affecting factors on the PSMP's complexity. The objective is to find out the advantages, disadvantages and limitations for each solution method for solving the problem under consideration. The comparison criteria will be solution quality, CPU time measured in seconds to reach best solutions and the consumed memory Megabytes consumed by the CPU during processing the various algorithms.

**Computational experiments:** We tested the proposed algorithms on a set of 100 test problems representing the full factorial design for three factors. These factors are system sizes of {5,10,20,30,50} machines, cycle lengths of {6,12,24,36,52} and number of different maintenance operations per cycle of {2,3,4,5}. We assumed that there are 9 maintenance operations' options where the first option represented an overhaul operation. The man-hours required for an overhaul is generated from a discrete uniform random number from [1500, 1600] while other operations man-hours is generated from the discrete uniform distributions from [45,55], [75,85], [85,100], [100,120], [120,140], [150,180], [180,200], [700,800] respectively. The operation's costs were generated from continuous uniform distributions from [1400,1500], [40,50], [70,80], [80,95], [95,115], [115,135], [145,170], [170,200] and [600,700]. A machine was selected as a standard machine randomly and machines' complexity parameters were generated from a continuous random number distribution from [0.5,2].





Table 2: Continue

72	20	24	3	2246421.3	1.0825	801	--	0.9541	35	--	0.1670	0.2378	0.3383	0.83	1	27	300/700
73	50	6	3	5120759.0	1.0563	747	--	0.4979	19	--	0.0832	0.0894	0.1022	0.75	1	50	350/800
74	5	24	3	2384947.1	1.0807	527	--	0.2883	13	--	0.2665	0.2754	0.3090	0.97	1	1	50/200
75	20	6	3	784981.0	1.0234	741	--	0.9074	10	23431	0.5423	0.5900	0.6620	0.87	1	43	400/1000
76	30	52	2	5269995.5	1.0892	983	--	0.8881	68	--	0.1394	0.2094	0.2910	0.77	2	23	150/800
77	30	24	2	2326750.5	1.0831	812	--	0.9978	54	--	0.1548	0.2023	0.2660	0.78	1	6	50/900
78	30	24	3	3378980.0	1.0834	798	--	0.8814	53	--	0.0961	0.1480	0.2431	0.77	1	30	250/800
79	20	52	5	2674943.5	1.0896	932	--	0.9769	71	--	0.1958	0.3237	0.4620	0.80	2	11	200/300
80	10	52	3	1200920.0	1.0887	913	--	0.9147	72	--	0.5429	0.5816	0.6404	0.89	2	6	100/500
81	50	36	5	8374066.0	1.0881	923	--	0.4962	51	--	0.0665	0.0968	0.1402	0.68	2	46	200/900
82	30	12	3	3348554.0	1.0721	769	--	0.9766	18	--	0.0342	0.1075	0.2567	0.78	1	26	250/800
83	30	24	4	4475539.5	1.0837	649	--	0.8615	54	--	0.0998	0.1522	0.2208	0.77	1	5	50/800
84	5	36	4	687471.1	1.0849	765	--	0.7811	29	--	0.7421	0.7749	0.9153	0.97	1	1	50/400
85	50	24	2	5642821.5	1.0842	893	--	0.9983	48	--	0.0576	0.0968	0.1565	0.73	1	18	100/900
86	20	24	4	4302062.0	1.0829	768	--	0.8426	36	--	0.1176	0.1563	0.2138	0.83	1	7	150/400
87	5	24	5	1499728.1	1.0806	439	--	0.5497	15	--	0.4550	0.4610	0.4851	0.98	1	1	50/400
88	20	12	5	2762268.5	1.0709	879	--	0.8384	20	--	0.0854	0.1411	0.2110	0.84	1	45	400/1000
89	5	52	4	461967.5	1.0888	678	--	0.8661	58	--	0.6226	0.6522	0.7335	0.96	1	1	50/300
90	20	52	4	3152566.0	1.0891	853	--	0.6999	72	--	0.1852	0.2348	0.2848	0.80	2	27	200/1000
91	10	52	4	1358620.6	1.0889	975	--	0.9207	73	--	0.4567	0.5167	0.5969	0.89	2	15	250/600
92	30	52	5	452829.5	1.0897	987	--	0.5694	68	--	0.1083	0.1644	0.2359	0.75	2	22	200/500
93	5	52	5	850346.3	1.0884	765	--	0.8545	62	--	0.6062	0.6530	0.7508	0.95	2	1	50/400
94	50	52	3	4834801.5	1.0897	987	--	0.9056	110	--	0.0800	0.1158	0.1646	0.66	2	56	200/900
95	50	36	4	5873769.5	1.0884	871	--	0.7163	50	--	0.0910	0.1246	0.1765	0.69	2	44	250/600
96	30	52	4	3752435.5	1.0896	956	--	0.9523	69	--	0.1590	0.2115	0.2784	0.75	2	37	200/1000
97	5	52	3	684493.8	1.0884	700	--	0.9600	60	--	0.8307	0.8446	0.9021	0.96	2	1	50/400
98	30	6	2	6159486.5	1.0409	765	--	0.5498	10	25432	0.0005	0.0060	0.0258	0.80	2	29	250/1000
99	30	24	5	3284380.0	1.0838	715	--	0.8871	56	--	0.1157	0.1745	0.2498	0.76	1	13	100/1000
100	20	12	4	2487208.3	1.0656	865	--	0.8245	19	--	0.1441	0.2008	0.2900	0.85	1	16	200/800
Average				2884336.7	1.0709	750	--	0.7958	37	--	0.2929	0.3367	0.4217	0.84	1	22	--

\* Indicates the solution is optimal, -- Indicates that the math programming stopping criterion of 36,000 sec

The MATLAB software was used to code the developed GA heuristic that 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. 100 test problems were generated for each design resulting 10000 instances. Each problem was solved by the mixed-integer non-linear and linearized math programming models using CPLEX software and hence solved by the developed GA. Table 2 summarizes the experimentation results. Columns 1-4 depict the design number, system size, cycle length and number of different maintenance operations. Columns 5 to 8 present the average lower bounds (LB) of the 100 problem per design, the average Upper Bounds (UB), the memory allocated to the linear solver and the time consumed in CPU seconds by CPLEX. Columns 8 to 10 give the performance of the mixed-integer non-linear programming model in terms of average lower bounds referenced to the lower bounds of the linear model given at column 5, the consumed memory and the CPU seconds. Columns 12-18 presents the GA's performance in terms of best, mean, worst, reliability (Rel) measured as the average number of times the GA could find the best solution divided by 100, memory consumed (Mem), CPU seconds and population size/number of generations. The "--" indicates that a math programming stopping criterion of 36,000 seconds. A random seed was used for each GA's evolution.

In general, the GA does not consume large memory bytes compared with the math models although there is a significant difference between the memory allocated for linear and nonlinear versions. The nonlinear version takes a smaller memory than the linear one due to the high number of extra constraints and binary decision variables of the linearization process. The GA consumes from 1-2 Megabytes only for each problem in the test set while the consumed memory could be 1 Gigabytes for some problems in the set.

For the smallest level of 5 machines and 6 planning periods, the nonlinear, linearized math models and the developed GA were able to identify the optimal solution but the GA solved each design in a second consuming Megabytes whereas math models consume larger CPU seconds and memory bytes. By increasing the number of machines, the developed algorithms could not define optimal solutions but GA could find solutions better than the math models for all remaining problems. The nonlinear math algorithm could find solutions with lower bounds better than the linearised version where the linear model defines feasible solutions for 4 problems. The nonlinear model trapped in local optimal solutions for 25 problems before achieving the stopping criteria of 36,000 seconds as listed in Table 2.

With regard to the CPU seconds, the developed GA could find better solutions rather than the math programming models where it takes on average 22 sec.

The mixed integer linear math programming model could not define solutions for 96 and 75 problems respectively. The GA success to define solutions with average reliability of 0.84. The Rel, CL and O were interrelated by the equation  $Rel = 1.02 - 0.00573 \times M - 0.00123 \times CL - 0.00388 \times O$  for the developed GA. The developed GA outperforms both the math programming models in finding best solutions in suitable computational time and Megabytes.

### CONCLUSION

This article addressed the minimization of a hard mixed-integer non-linear optimization problem related to the periodic maintenance scheduling. The problem had a predefined sequence of maintenance operations on a set of machines for a definite planning period. The objective was to minimize the total man-hours and cost variances along the planning horizon with equal weights. We developed a mixed-integer non-linear math programming model and the linearised version of the problem. Also, we developed a GA for solving the problem and compared the performance of algorithms on a generated set of test problems. The results indicated the superiority of the developed GA on the math programming models with respect to solution quality, memory allocated to each algorithm and the CPU seconds.

The natural expansion to this work is to consider the relevant and efficient meta-heuristics like simulated annealing, taboo search, particle swarm and ant colony optimization approaches and analyzing their performances in order to get a better understanding of how benefits are attained from using each approach. Also, we can consider the integration between maintenance planning and aggregate production planning problems. A possible work could be done by considering other maintenance strategies as reliability-based and condition-based approaches. Analyzing the resulting linear integer programming model for figuring out integrality gaps and valid inequalities strengthening the formulation would be a nice expansion to the field.

### REFERENCES

Anily, S., C.A. Glass and R. Hassin, 1999. Scheduling of maintenance services to three machines. *Annals Operat. Res.*, 86: 375-391. DOI: 10.1023/A:1018971222185

Anily, S., C.A. Glass and R. Hassin, 1998. The scheduling of maintenance service. *Discrete Applied Math.*, 82: 27-42. DOI: 10.1016/S0166-218x(97)00119-4

Bar-Noy, A., R. Bhatia, J.S. Naor and B. Schieber, 2002. Minimizing service and operation costs of periodic scheduling. *Math. Operat. Res.*, 27: 518-544. DOI: 10.1287/moor.27.3.518.314

Beasley, J.E. and P.C. Chu, 1996. A genetic algorithm for the set covering problem. *Eur. J. Operat. Res.*, 94: 392-404. DOI: 10.1016/0377-2217(95)00159-X

Brauner, N., Y. Crama, A. Grigoriev and J.J. van de Klundert, 2001. On the complexity of high multiplicity scheduling problems. <http://hal.archives-ouvertes.fr/hal-00083365/>

Charest, M. and J.A. Ferland, 1993. Preventive maintenance scheduling of power generating units. *Annals Operat. Res.*, 41: 185-206. DOI: 10.1007/BF02023074

Davis, L., 1991. *Handbook on Genetic Algorithms*. 1st Edn., Van Nostrand Reinhold, New York, ISBN: 0442001738, pp: 385.

Dekker, R., 1996. Application of maintenance optimization models: A review and analysis. *Reliability Eng. Syst. Safety*, 51: 229-240. DOI: 10.1016/0951-8320(95)00076-3

Dekker, R., F.A. van der Duyn Schouten and R.E. Wildeman, 1997. A review of multi-component maintenance models with economic dependence. *Math. Methods Operat. Res.*, 45: 411-435. DOI: 10.1007/BF01194788

Duffuaa, S.O. and M. Ben-Daya, 1994. An extended model for the joint overhaul scheduling problem. *Int. J. Operat. Prod. Manage.*, 14: 37-43. DOI: 0.1108/01443579410062158

Gertsbakh, I. and E. Gertsbakh, 2000. *Reliability Theory with Applications to Preventive Maintenance*. 1st Edn., Springer-Verlag, Heidelberg, ISBN3540672753., pp: 219.

Goldberg, D.E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st Edn., Addison-Wesley, Reading, MA., ISBN: 0201157675, pp: 412.

Grigoriev, A., J. Joris van de Klundert and F. Spieksma, 2006. Modeling and solving the periodic maintenance problem. *Eur. J. Operat. Res.*, 172: 783-797. DOI: 10.1016/J.EJOR.2004.11.013

Hariga, M., 1994. A deterministic maintenance-scheduling problem for a group of non-identical machines. *Int. J. Operat. Prod. Manage.*, 14: 27-36. DOI: 10.1108/01443579410062149

Kenyon, C. and N. Schabanel, 2001. The Data Broadcast Problem with Non-Uniform Transmission Times. Research Report 2001-43, Laboratoire de l'Informatique du Parallélisme, École Normale Supérieure de Lyon. <http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.22.2061>

- Kenyon, C., N. Schabanel and N. Young, 2000. Polynomial-time approximation scheme for data broadcast. Proceedings of the 32nd ACM Symposium on Theory of Computing, May 21-23, Portland, OR, USA., pp: 659-666. DOI: 10.1145/335305.335398
- Lorena, L.A.N. and L.S. Lopes, 1997. Genetic algorithms applied to computationally difficult set covering problems. *J. Operat. Res. Society*, 48: 440-445. DOI: 10.1057/palgrave.jors.2600380
- 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
- Michalewicz, Z., 1996. *Genetic Algorithms + Data Structure = Evolution Programs*. 3rd Edn., Springer Verlag, New York, ISBN: 3540606769, pp: 387.
- Schabanel, N., 2000. The databroadcast problem with preemption. Proceedings of the 17th International Symposium on Theoretical Aspects of Computer Science (STACS'00), Springer-Verlag London, UK., pp: 181-192. ISBN: 3-540-67141-2
- Sherif, Y.S. and M.L. Smith, 1981. Optimal maintenance models for systems subject to failure—a review. *Naval Res. Logistics Quar.*, 28: 47-74. DOI: 10.1002/nav.3800280104
- Sule, D.R. and B. Harmon, 1979. Determination of coordinated maintenance scheduling frequencies for a group of machines. *AIIE Trans.*, 11: 48-53. <http://www.tandfonline.com/doi/abs/10.1080/05695557908974400#preview>
- Wagner, H.M., R.J. Giglio and R.G. Glaser, 1964. Preventive maintenance scheduling by mathematical programming. *Manage. Sci.*, 10: 316-334. DOI: 10.1287/mnsc.10.2.316