

ACCEPTED MANUSCRIPT

This is an Accepted Manuscript of an article published by Taylor & Francis Group in *International Journal of Production Research* on 09/06/2013, available online at:

<http://www.tandfonline.com/doi/full/10.1080/00207543.2013.784411>



Using response surface design to determine the optimal parameters of genetic algorithm and a case study

Ibrahim KUCUKKOC^{a,b}, Aslan Deniz KARAOGLAN^b, Ramazan YAMAN^b

^aUniversity of Exeter, College of Engineering Mathematics and Physical Sciences, Exeter, United Kingdom

^bBalikesir University, Department of Industrial Engineering, Balikesir, Turkey
i.kucukkoc@exeter.ac.uk, deniz@balikesir.edu.tr, ryaman@balikesir.edu.tr

Abstract: Genetic algorithms are efficient stochastic search techniques for approximating optimal solutions within complex search spaces and used widely to solve NP hard problems. This algorithm includes a number of parameters whose different levels affect the performance of the algorithm strictly. The general approach to determine the appropriate parameter combination of genetic algorithm depends on too many trials of different combinations and the best one of the combinations that produces good results is selected for the program that would be used for problem solving. A few researchers studied on parameter optimisation of genetic algorithm. In this paper, response surface depended parameter optimisation is proposed to determine the optimal parameters of genetic algorithm. Results are tested for benchmark problems that is most common in mixed-model assembly line balancing problems of type-I (MMALBP-I).

Keywords: Genetic algorithm (GA), Response surface methodology (RSM), Assembly line balancing, Parameter optimisation, Design of experiment.

Corresponding author: Ibrahim Kucukkoc, I.Kucukkoc@exeter.ac.uk

Using response surface design to determine the optimal parameters of genetic algorithm and a case study

Genetic algorithms are efficient stochastic search techniques for approximating optimal solutions within complex search spaces and used widely to solve NP hard problems. This algorithm includes a number of parameters whose different levels affect the performance of the algorithm strictly. The general approach to determine the appropriate parameter combination of genetic algorithm depends on too many trials of different combinations and the best one of the combinations that produces good results is selected for the program that would be used for problem solving. A few researchers studied on parameter optimisation of genetic algorithm. In this paper, response surface depended parameter optimisation is proposed to determine the optimal parameters of genetic algorithm. Results are tested for benchmark problems that is most common in mixed-model assembly line balancing problems of type-I (MMALBP-I).

Keywords: Genetic algorithm (GA), Response surface methodology (RSM), Assembly line balancing, Parameter optimisation, Design of experiment.

1. Introduction

Genetic algorithms (GAs) are part of the so-called evolutionary algorithms based on natural genetics that provide robust search capabilities in complex spaces, and they thereby offer a valid approach to problems requiring efficient and effective search processes. The basic idea is to maintain a population of individuals those representing candidate solutions to the problem that evolves over time through a process competition and they do not need to have any information about the search space, just needing a fitness function that assigns a value to each solution (Costa *et al.*, 2005; Fernandez-Prieto *et al.*, 2011).

The running principle of GA depends on setting up a relatively large number of parameters and requires performing lots of runs for different combinations of them to obtain the most appropriate structure that gives nearly optimal solution. There are few

researches about parameter optimisation of GAs. Lobo and Goldberg (2004) presented the *parameter-less genetic algorithm* to simplify genetic algorithm operation by incorporating knowledge of parameter selection and population sizing theory in the genetic algorithm and applied it to a network expansion problem. Siriwardene and Perera (2006) used *proportionate* and *linear ranking* selection methods to determine model parameter convergence and recommended the proportionate selection method for urban drainage model parameter optimisation. Fernandez-Prieto *et al.* (2011) used fuzzy logic to determine GA control parameters (fuzzy adaptive genetic algorithms – FAGAs).

Taguchi experimental design is one of the well-known and effective design of experiment techniques that requires less experiment and used to find the best combination of parameters. Yang *et al.* (2005) combined the Taguchi experimental method with the GA to find the best combination of the GA parameters. One disadvantage of Taguchi method is that it does not give the mathematical model of the relations between parameters so predicting the intermediate values is impossible. Subbaraj *et al.* (2011) proposed a new optimisation algorithm, namely Taguchi self-adaptive real-coded genetic algorithm (TSARGA) and implemented to solve economic dispatch (ED) problem with valve-point loading. Chang (2011) used Taguchi method, in order to determine the important parameters in genetic algorithm neural networks (GANN) with the goal of reducing the estimation error. Also Factorial design, which is one of the widely used design of experiment techniques, is used to model the non-quadratic relations between the factors and the response. The third and the widely used well-known design of experiment technique is the response surface methodology (RSM) which is based on statistical considerations that brings the most meaningful information about the influences of parameters on a specific problem; and process

optimisation using RSM is usually achieved by simultaneous testing of numerous factors in a limited number of experiments. Therefore, RSM consumes less time and effort (Bayhan and Onel, 2010; Costa *et al.*, 2005). Remarkable studies in the literature on this subject begin with Pongcharoen *et al.* (2002). In this study, the authors performed a *factorial experiment* to identify appropriate values of GA parameters that produce the best results within a given execution time and developed a genetic algorithm-based scheduling tool (GAST) for scheduling of complex products with multiple resource constraints and deep product structure. Costa *et al.* (2005) applied fractional factorial design for the selection of the parameters of GA and applied GA to a batch cooling crystallisation optimisation. The brief literature review is summarised in Table 1.

(Table 1. An overview of approaches in the literature on parameter optimisation of GAs.)

According to the Table 1, it is clearly observed that the studies on determining optimal parameter combination of GAs are limited and response surface optimisation is not used for determining optimal GA parameters. In the present study mixed-model assembly line balancing problem of type-I is handled for a benchmark problem (test problem-8) taken from Simaria (2006) and the regression equation and optimum parameters calculated by response surface optimiser of Minitab-16 package is presented. Then the experimental study is extended for the benchmark problems. When we reviewed the literature, we observed that RSM is not used for this purpose. The originality of this work is that applying RSM to the parameter optimisation of GA for MMALBP-I problems.

The rest of the paper is organised as follows. The concept and formulations of GA, RSM and MMALBP-I are presented in section 2. An illustrative example is solved in

section 3. Discussions are given with the obtained results of test problems in section 4 and conclusion are given in section 5 with future research directions.

2. Proposed genetic algorithm

Genetic algorithms (GAs) are powerful and broadly applicable random search and optimisation techniques based on principles of evolution theory. In recent years, GAs have been known to be extremely efficient technique for solving NP Hard problems (Zaman *et al.*, 2012). The following subsections describe in detail the features of the GA which is used in this research.

In the algorithm, we used task based representation which was used by Leu *et al.* (1994), Sabuncuoglu and Tanyer (2000), Akpinar and Bayhan (2011) that is the most appropriate chromosome type for line balancing problems. The length of the chromosome is defined by adding two genes to the number of tasks. These two genes represent the station number and fitness value (FV) at the end of each chromosome. In task based sequences, the chromosomes need to fit the precedence relationship diagram because of the technological precedence restrictions. Therefore, permutation coding method was used to sequence the tasks on the chromosomes without any repetition.

Random initialisation method that initialises each chromosome randomly was used to produce starting population. Initialisation method was specified in order to prevent unsuitable chromosomes, which interrupt precedence relationships.

GA used in this research employs the objective function given in Equation (1) as the fitness function to evaluate each individual's performance (fitness) in the search space.

$$\text{Fitness Value} = \sum_{k=1}^S \frac{(C - W_k)^2}{S} \quad (1)$$

where C , W_k and S denote the cycle time of the assembly line, workload of the station k and the number of workstations required to meet the demand in the assembly line, respectively.

Individuals are selected randomly for mating. Thus, diversity is to be protected and probable better solutions can be occurred with the mating of worse individuals.

Two-point crossover method was used to perform the crossover operator according to the predefined crossover rate (C_R). For mating, the individuals are selected randomly and paired with the other individuals.

The mutation operator exchanges randomly selected two genes according to the mutation rate (M_R) in a chromosome that is selected randomly. After the crossover and mutation operators, FVs of the new offspring are compared with the individuals' and the ones that have better FVs are transferred to next generation.

Elitism operator is applied indirectly by ensuring that none of the best individuals are missed during the iterations.

New generation is formed by selecting the best individuals according to their FVs among the current population, offspring produced by crossover, and individuals who underwent mutation. Population size is kept constant by replacing the worse individuals with new ones by taking into account the FVs.

Repairing operator is not used in this algorithm since unsuitable chromosomes are not allowed to be produced during the initialisation, crossover and mutation processes.

Figure 1 shows the flow chart of GA used in this research.

(Figure 1. Flow chart of proposed genetic algorithm.)

2.1. Assembly line balancing problem

“Assembly Line Balancing Problem (ALBP) was firstly formulated by Salveson (1955) and, since then, has received a great deal of attention over the years” (Hamzadayi and Yildiz, 2012). An assembly line consists of a number of workstations, linked together with a material handling system such as conveyor or moving belt (Azzi *et al.*, 2012). Most common and difficult problem is to determine how these tasks can be assigned to the stations fulfilling certain restrictions since the manufacturing process is divided into a set of tasks. So, the main objective is to determine optimal assignment of the tasks to the workstations (Chica *et al.*, 2011).

There are three types of mixed-model assembly line balancing problems in the literature and these are classified according to their objective functions. The types of the MMALBP are (Scholl, 1995):

- MMALBP-I: The number of workstations is to be minimised for a given cycle time.
- MMALBP-II: The cycle time is to be minimised for a given number of workstations.
- MMALBP-E: The cycle time and the number of workstations are to be minimised simultaneously.

The objective function used as fitness function of genetic algorithm (Equation 1) was taken from Leu *et al.* (1994). This function minimises not only total number of workstations but also workload smoothing between the workstations at the same time.

The decision variables x_{ik} and r_k check whether task i is assigned to workstation k and if workstation k is replicated or not, respectively.

$$x_{ik} = \begin{cases} 1, & \text{if task } i \text{ is assigned to workstation } k, \\ 0, & \text{otherwise} \end{cases}$$

$$r_k = \begin{cases} 1, & \text{if workstation } k \text{ is replicated,} \\ 0, & \text{otherwise} \end{cases}$$

Assumptions of the model are (Akpınar and Bayhan, 2011; Vilarinho and Simaria, 2002):

- the planning horizon has a fixed length P ,
- a set of similar M models can be simultaneously assembled,
- the forecast demand, over the planning horizon, for model m is D_m , requiring the line to be operated with a cycle time;

$$C = P / \sum_{m=1}^M D_m \quad (2)$$

- the overall proportion of the number of units of model m being assembled is then;

$$q_m = D_m / \sum_{p=1}^M D_p \quad (3)$$

(where D_p describes amount of total production for all models),

- each model has its own set of precedence relationships, but there is a subset of tasks common to all models. Hence, the precedence diagrams for all the models can be combined and the resulting one has N tasks,
- N tasks are performed in a set of S workstations,
- the time required to perform task i on model m , t_{im} , may vary among models ($t_{im} = 0$ means that model m does not require task i to be assembled),
- a task can be assigned to only one workstation and, consequently, the tasks that are common to several models need to be performed on the same workstation,
- the set of tasks that cannot be performed before task i is completed, F_i (successors of task i), is given by the precedence constraints derived from the combined precedence diagram,
- the zoning constraints are defined in the assembly process, ZP is the set of task pairs that must be assigned to the same workstation (compatible tasks) and ZN is the set of task pairs that cannot be performed at the same workstation (incompatible tasks),
- a workstation can be duplicated up to a maximum of $MAXP$ replicas, but only if the task time of one of the tasks assigned to it exceeds a predefined value ($\alpha\%$ of the cycle time) for at least one of the models (for our model $MAXP = 2$),
- S_{km} symbolises the idle time of workstation k for model m .

Constraints considered in the model are (Vilarinho and Simaria, 2002):

$$\sum_{k=1}^S x_{ik} = 1 \quad i = 1, \dots, N \quad (4)$$

$$\sum_{k=1}^S x_{ak} - \sum_{k=1}^S x_{bk} \leq 0 \quad a \in N, b \in F_a \quad (5)$$

$$\sum_{k=1}^S x_{ak} - \sum_{k=1}^S x_{bk} = 0 \quad (a, b) \in ZP \quad (6)$$

$$x_{ak} + x_{bk} \leq 1 \quad (a, b) \in ZN, k = 1, \dots, S \quad (7)$$

$$\sum_{i=1}^N t_{im} x_{ik} + S_{km} = C [1 + r_k (MAXP - 1)] \quad k = 1, \dots, S, m = 1, \dots, M \quad (8a)$$

$$r_k \leq \sum_{i: \exists t_{im} > \alpha C; m=1, \dots, M} x_{ik} \quad k = 1, \dots, S, 0 < \alpha < 100\% \quad (8b)$$

$$Mr_k \geq \sum_{i: \exists t_{im} > \alpha C; m=1, \dots, M} x_{ik} \quad k = 1, \dots, S, 0 < \alpha < 100\% \quad (8c)$$

$$S_{km} \geq 0 \quad k = 1, \dots, S, m = 1, \dots, M \quad (9a)$$

$$x_{ik} \in [0, 1] \quad k = 1, \dots, S, i = 1, \dots, N \quad (9b)$$

$$r_k \in [0, 1] \quad k = 1, \dots, S \quad (9c)$$

The objective function (1) minimises the sum of the squares of the idle times for each workstation, thus it minimises not only total number of the workstations but also unbalanced workload between the workstations. Constraint (4) ensures that each task is assigned to exactly one workstation. Constraint (5) ensures none of the successors of a task is assigned to an earlier station than that task. Constraint (6) describes positive zoning constraints (compatibility zoning constraint). For instance, if any two tasks need to be performed at same workstation because of the technological restrictions, they must be assigned to same workstation. Constraint (7) describes negative zoning constraints (incompatibility zoning constraint). For example if it is dangerous or impossible to perform any two tasks at same workstation, they must be assigned to different workstations. Constraint (8a) ensures that capacity is not exceeded for any workstation, constraint (8b) ensures the maximum number of replicas of a workstation is not exceeded and constraint (8c) ensures only workstations where the processing time of the

tasks assigned to it, for at least one model, exceeds a certain proportion ($\alpha\%$) of the cycle time can be duplicated (where M is a very large positive integer). Constraint (9a) describes the idle time of workstations equal or more than zero. Additionally, constraints (9b) and (9c) define domain of the decision variables (Akpinar and Bayhan 2011; Vilarinho and Simaria, 2002).

2.2. Response surface methodology

RSM is a union of statistical and mathematical techniques used for modelling the mathematical relations between the inputs and outputs of a process which is necessary for developing, improving, and optimising processes. RSM has been used extensively in the engineering problems to examine and characterise problems in which input variables influence some performance aspects of the product or process. This performance measure is called as response. Product or process optimisation using RSM is usually achieved by simultaneously testing of numerous factors (controllable input variables) in a limited number of experiments. Therefore, RSM consumes less time and effort. Furthermore, RSM provides quantitative measurements of possible interactions between factors, difficult information to obtain using other optimisation techniques (especially by using heuristics). Detection and quantification of the interactions between various factors are important at the optimisation stage in engineering problems (Bayhan and Onel, 2010). RSM was proposed by Box and Wilson (1951) for finding the input combination that minimises the output of a real non-simulated system. In most RSM problems, the form of the relationship between independent variables and the response is unknown, and approximated (Dhupal *et al.*, 2007). Equation (10) shows the general second-order polynomial response surface mathematical model (full quadratic model) for the experimental design (Yalcinkaya and Bayhan, 2009, Dhupal *et al.*, 2007).

$$Y_u = \beta_0 + \sum_{i=1}^n \beta_i X_{iu} + \sum_{i=1}^n \beta_{ii} X_{iu}^2 + \sum_{i < j}^n \beta_{ij} X_{iu} X_{ju} + e_u \quad (10)$$

where Y_u is the corresponding response, X_{iu} and X_{ju} are coded values of the i th and j th input parameters ($i < j$) respectively, terms β_0 , β_i , β_{ii} and β_{ij} are the regression coefficients, and e_u is the residual experimental error of the u th observation.

The model in terms of the observations may be written in matrix notation as (Montgomery, 2001):

$$Y = \beta X + \varepsilon \quad (11)$$

where Y is the output matrix and X is the input matrix, and ε is the matrix of residuals (error term). The least square estimator of β matrix that composes of coefficients of the regression equation calculated by the given formula in Equation (12):

$$\beta = (X'X)^{-1} X'Y \quad (12)$$

The fitted regression models, with the coefficients for FV are given in the next section.

3. Results

3.1. Parameter optimisation of GA

Randomised experimental runs were carried out to minimise the error. The factor levels of genetic algorithm parameters for the experiments are generation number (G_N), population size (P_S), crossover rate (C_R) and mutation rate (M_R) are listed in Table 2.

The levels of these parameters are determined randomly by considering the levels from similar studies presented previously in the literature based on GA and MMALBP-I. Table 3 shows the experimental design, detailing the experiment run order of each experiment and coded values of the process parameters (namely factors). For each

experiment 100 runs are performed. Minitab-16 statistical software was used to establish mathematical models for optimisation of the GA parameters. The software uses the mathematical substructure given in Section 2.2.

(Table 2. Levels and values of genetic algorithm parameters G_N , P_S , C_R , and M_R .)

(Table 3. Design of experiments matrix showing coded values and observed responses for 100 runs.)

Mathematical model based on RSM for correlating responses such as the parameters G_N , P_S , C_R and M_R with various settings of the process parameters as considered in the experimental design have been established, and is represented in the Equation (13).

$$\begin{aligned}
 FV = & 10.0788 - 0.23994G_N - 0.43357P_S - 0.12353C_R - 0.09101M_R + 0.174854G_N^2 + 0.225179P_S^2 \\
 & + 0.116204C_R^2 + 0.010491M_R^2 - 0.12087G_NP_S + 0.090719G_NC_R - 0.13349G_NM_R \\
 & - 0.03798P_SC_R - 0.02787P_SM_R - 0.02046C_RM_R
 \end{aligned}
 \tag{13}$$

Optimisation procedure was performed to achieve the target value of fitness function (FV) that is given in Equation (1). Optimum parameter settings obtained are presented in Figure 2.

(Figure 2. Optimisation results for G_N , P_S , C_R , and M_R .)

The current coded optimal process parameter settings for achieving the targeted fitness value of $FV = 9$ are calculated as $G_N = 1.9596$, $P_S = 1.6364$, $C_R = 0.2222$, and $M_R = 2$.

After the normalisation, the uncoded values are calculated as $G_N = 1039.9$, $P_S = 92.728$, $C_R = 0.5444$ and $M_R = 0.25$ respectively (Table 4). The

response is optimised at the above parametric combination with desirability (d) of 0.9912 (99.12%) and optimised response value of FV is calculated as 9.0264.

(Table 4. Results of parameter optimisation.)

The experimental results for the given case under this study are presented in the next section.

3.2. Experimental results for MMALBP-I

The GA was tested on a benchmark problem (test problem 8) given in Table 5 with the parameters found by response surface methodology. The main characteristics of the most common test problems in the relevant literature are presented in Table 5 and the compared results with the previous results demonstrated in the literature for test problem 8 are shown in Table 6. To describe the problems, the number of tasks of the combined precedence diagram (N), the number of models (M) and the cycle time of the assembly line (C) are given in the columns three to five and nine to eleven in Table 5.

(Table 5. Test problems (Akpınar and Bayhan, 2011; Scholl, 1995; Vilarinho and Simaria, 2002).)

The precedence diagram (Figure 3) and task processing times used for the test problem 8 (with 25 Tasks, 3 models) was taken from Simaria (2006). The coding of the genetic algorithm was developed in Matlab R2008a and implemented on Intel Core i5 CPU M480 2.67 GHz system with 3 GB RAM and 64 Bit operating system.

(Figure 3. Precedence diagram of test problem 8 (Simaria, 2006).)

The computational results for the given test problem are presented in Table 6. LB_{pmix} column shows the lower bound on the total number of workstations for relevant problem (Vilarinho and Simaria, 2002). The columns Kilbridge & Wester, Moodie &

Young, RPWT, Pure GA (Akpinar and Bayhan, 2011) show the best solutions in the relevant literature found by different solution approaches. V&S (SA) and A&B (hGA) columns represent the best solutions found by Vilarinho and Simaria (2002) by using simulated annealing (SA) algorithm, and Akpinar and Bayhan (2011) by using hybrid genetic algorithm (hGA), respectively. RSM-GA column represents the best station number found by using optimised parameters of GA obtained from RSM. The task sequences discovered by RSM-GA procedure are given in appendices.

(Table 6. Computational results (station numbers) for test problem 8.)

3.3. Comparisons for well-known design of experiment techniques

In this section a comprehensive experimental design is performed to clearly show the advantage of using RSM instead of using other well-known design of experiment techniques. For this purpose experimental designs are performed for the two well-known design of experiment technique namely Taguchi Method and 2^k Factorial Design.

As is well known in the literature that Taguchi method requires less experiment when compared with RSM but this method provides only the optimum combination of factor levels. So RSM is more appropriate method for this type of problems because it is possible with RSM to calculate the optimum factor levels with decimals as can be seen in Table 4. In other words it is possible to obtain more sensitive parameter levels with RSM when it is compared with the solutions of Taguchi Method. The same sensitivity may also be obtained by 2^k factorial design but this technique depends on the first order (linear) relations between the independent factors. The parameter optimisation case presented in this study shows nonlinear relations because of the nature of the problem. So 2^k factorial design does not seem a good alternative to the RSM. The factor levels for

Taguchi Method and 2^k Factorial Design are presented in Tables 7 and 10, and the simulation results for the given experimental designs are presented in Tables 8 and 11 respectively.

(Table 7. Factor levels and values of genetic algorithm parameters for Taguchi design.)

By using the five level design of Taguchi for four factors the design composed of 25 experiments is obtained and displayed in Table 8 and the average FV values are calculated for each parameter combination.

(Table 8. Taguchi design and simulation results for average FV.)

For the optimisation, smaller-is-best criterion of Taguchi method is used. For minimising average FV; Signal to Noise (S/N) ratios for each experimental runs are calculated from Minitab. Minitab uses the formula given below in Equation (14) to calculate S/N ratios for “smaller is better” criteria:

$$S/N = -10(\log(\sum Y^2/n)) \quad (14)$$

Calculated signal to noise (S/N) ratios for smaller is better criteria are listed in Table 9. The factor levels that have the maximum S/N ratio for each factor is selected as the optimum factor level for the factors and given in bold.

(Table 9. Signal to noise (S/N) ratios for smaller is better criteria.)

According to the S/N ratios optimum coded parameter combination is found as $G_N(2)$, $P_S(5)$, $C_R(1)$, and $M_R(2)$. The uncoded levels are calculated as $G_N(300)$, $P_S(100)$, $C_R(0.1)$, and $M_R(0.1)$. Another well-known design of experiment technique is the 2^k factorial design as mentioned previously. The factor levels for 2^k design are determined as given in Table 10.

(Table 10. Factor levels and values of genetic algorithm parameters for 2^k factorial design.)

By using the two level factorial design for four factors the design composed of 16 experiments is obtained and displayed in Table 11 and the average FV values are calculated for each parameter combination.

(Table 11. 2^k factorial design and simulation results for average FV.)

Mathematical model based on 2^k factorial design for correlating responses such as the parameters G_N, P_S, C_R and M_R with various settings of the process parameters as considered in the experimental design have been established, and is represented in the Equation (15).

$$\begin{aligned}
 FV = & 11.3563 - 0.1021 G_N - 1.0813 P_S + 0.2306 C_R - 0.3459 M_R + 0.5088 G_N P_S \\
 & + 0.3500 G_N C_R - 0.0088 G_N M_R + 0.0603 P_S C_R - 0.0084 P_S M_R \\
 & + 0.3161 C_R M_R - 0.2394 G_N P_S C_R - 0.3581 G_N P_S M_R \\
 & + 0.2810 G_N C_R M_R - 0.235 P_S C_R M_R \\
 & - 0.4508 G_N P_S C_R M_R
 \end{aligned}
 \tag{15}$$

Optimisation analysis was performed to achieve the target value of fitness function (FV) that is given in Equation (1). Optimum parameter settings obtained are presented in Figure 4.

(Figure 4. Optimisation results for the G_N, P_S, C_R and M_R for 2^k factorial design.)

The current coded optimal process parameter settings for achieving the targeted fitness value of $FV = 9$ are $G_N(-1), P_S(1), C_R(-1)$, and $M_R(1)$. The uncoded values are calculated as $G_N(50), P_S(100), C_R(0.1)$ and $M_R(0.25)$ respectively. The response is optimised at the above parametric combination with desirability (d) of 0.5508 (55.08%) and optimised response value of FV is calculated as 9.4492.

The computational results (station numbers) for test problem 8 with the optimum parameter levels are calculated as 15 stations for both Taguchi Method and 2^k factorial design. Results indicate that RSM is more efficient with 14 stations as mentioned before in Table 6.

4. Discussion

As can be seen from Table 6, GA with proposed parameters calculated by using RSM has found 14 work stations for test problem 8 and this result is equal or better than previous works for medium sized benchmark problem of Vilarinho and Simaria (2002). Furthermore, for the mentioned test problem, obtained solutions by proposed method in this paper are equal to the theoretical lower bound of total workstations (LB_{pmix}). Results demonstrate that without using any hybridisation, obtained solution by only optimising the parameters of pure GA with RSM, is better than the solutions of Kilbridge & Wester, Moodie & Young (Phase I), RPWT, Pure GA, and simulated annealing results (SA) of Vilarinho and Simaria (2002). Additionally, obtained result equals to the result that obtained from hybrid GA of Akpinar and Bayhan (2011). The proposed GA parameters which are given in Table 4 may be used as the best parameters set for problem 8 for the future researches.

Similarly, further experiments are carried out to solve other MMALBP-I test problems in Table 5 using proposed procedures with optimised parameters. The obtained total number of required station numbers for each test problem are given in the RSM-GA column in Table 12.

(Table 12. Comparison of obtained results (station numbers) for test problems.)

Based on the results demonstrated in Table 12, it is clearly seen that proposed method outperforms three traditional heuristics (Kilbridge & Wester, Moodie & Young - Phase I, RPWT), and Pure GA. The proposed method produces better solutions than Pure GA for five test problems P8, P9, P16, P18, and P19. Moreover, equal solutions are observed for remaining test problems (P1-P7, P10-P15, P17, and P20). Hence, it can be said that GA may have a potential capacity to generate better solutions than Pure GA by only optimising its parameters without using any hybridisation.

In terms of comparison between simulated annealing algorithm (Vilarinho and Simaria, 2002) and RSM-GA, better solutions are obtained for three test problems (P8, P9, and P18) while in tie for remaining problems except P20. Furthermore, found solutions are equal to hGA (Akpınar and Bayhan, 2011) for 18 test problems while worse than hGA only for P15 and P20.

To conclude, the present paper demonstrates that by using only RSM, it is possible to obtain better results. The advantage of using RSM in the present study is obtaining the acceptable results with only running 31 experiments with 100 runs (preferable number of runs may be lower than 100 if the CPU time is quite high) for the given case without using any hybridisation.

While RSM appears to be quite useful for analysing many simulation problems, it has not received much attention from practitioners, despite efforts to encourage its application (Safizadeh and Thornton, 1984). The results of this study showed one more time that the RSM is an efficient statistical technique necessary for developing, improving, and optimising processes. The application of RSM is quite easy when compared with the other hybridisations mentioned in the literature. This may provide convenience to the researchers.

5. Conclusion and future research

The aim of this paper is to search the effect of parameter optimisation on GAs' performance. For this purpose, response surface methodology is applied to average fitness values of 100 replicates of different parameter combinations for the test problem of Vilarinho and Simaria (2002) (Problem 8, 25 Tasks, 3 Models, Medium-sized problem of MMALBP-I) and best parameter combination is tried to find. By using the optimal parameters of GA calculated by response optimiser, it was tested for the mentioned problem several times and 14 workstations with alternative task sequences are obtained. The objective was minimisation of number of workstations. By the proposed method, the probability of finding good solutions was increased by 6.67% $\left(\frac{15-14}{15}=0.067\right)$ for the 8th test problem. Furthermore, a comprehensive experimental design is performed to see the advantages of RSM by comparing the results obtained from RSM with the Taguchi and 2^k factorial design methods. Then to demonstrate the performance of the RSM-GA, 20 test problems from the literature (given in Table 5) are solved by using RSM-GA and results are compared with existing results in the relevant literature. The results presented in this study obviously show that the proposed approach based on RSM can successfully be used for the fast and easy design of the optimal parameter levels of GA for MMALBP-I. Also according to the results it is clearly observed that the solution quality of the traditional GA was improved.

Acknowledgments

The authors would gratefully like to thank to the editor and the anonymous referees whose valuable suggestions lead to an improved organisation of this paper.

References

- Akpınar, S., Bayhan, G.M., 2011. A hybrid genetic algorithm for mixed model assembly line balancing problem with parallel workstations and zoning constraints. *Engineering Applications of Artificial Intelligence*, 24, 449-457.
- Azzi, A., Battini, D., Faccio, M., Persona, A., 2012. Mixed model assembly system with multiple secondary feeder lines: layout design and balancing procedure for ATO environment. *International Journal of Production Research*, 50, 5132-5151.
- Bayhan, M., Onel, K., 2010. Optimization of reinforcement content and sliding distance for AlSi7Mg/SiCp composites using response surface methodology. *Materials and Design*, 31, 3015-3022.
- Box, G.E.P., Wilson, K.B., 1951. On the experimental attainment of optimum conditions. *Journal of Royal Statistical Society Series, B* 13, 1-38.
- Chang, K. Y., 2011. The optimal design for PEMFC modeling based on Taguchi method and genetic algorithm neural networks. *International Journal of Hydrogen Energy*, 36, 13683-13694.
- Chica, M., Cordon, O., Damas, S., Bautista, J., 2011. Multiobjective memetic algorithms for time and space assembly line balancing. *Engineering Applications of Artificial Intelligence*, doi:10.1016/j.engappai.2011.05.001.
- Costa, C. B. B., Maciel, M. R. F., Filho, R. M., 2005. Factorial design technique applied to genetic algorithm parameters in a batch cooling crystallization optimization. *Computers & Chemical Engineering*, 29, 2229-2241.
- Dhupal, D., Doloi, B., and Bhattacharyya, B., 2007. Optimization of process parameters of Nd:YAG laser microgrooving of Al₂TiO₅ ceramic material by response surface methodology and artificial neural network algorithm. *Proceedings of the Institution of Mechanical Engineers Part B-Journal of Engineering Manufacture*, 221, 1341-1351.
- Fernandez-Prieto, J. A., Canada-Bago, J., Gadeo-Martos, M. A., Velasco, Juan R., 2011. Optimisation of control parameters for genetic algorithms to test computer networks under realistic traffic loads. *Applied Soft Computing*, 11, 3744-3752.
- Hamzadayi, A., Yildiz, G., 2012. A genetic algorithm based approach for simultaneously balancing and sequencing of mixed-model U-lines with parallel workstations and zoning constraints. *Computers & Industrial Engineering*, 62, 206-215.

- Leu, Y. Y., Matheson, L. A., Rees, L. P., 1994. Assembly Line Balancing Using Genetic Algorithms with Heuristic-Generated Initial Populations and Multiple Evaluation Criteria. *Decision Sciences*, 25, 581–605.
- Lobo, F. G., Goldberg, D. E., 2004. The parameter-less genetic algorithm in practise. *Information Sciences*, 167, 217–232.
- Montgomery, D.C., 2001. Design and Analysis of Experiments, 5th Edition, John Wiley & Sons, Inc., New York.
- Pongcharoen, P., Hicks, C., Braiden, P.M., Stewardson, D.J., 2002. Determining optimum Genetic Algorithm parameters for scheduling the manufacturing and assembly of complex products, *International Journal of Production Economics*, 78, 311-322.
- Sabuncuoglu, E. E., Tanyer, M., 2000. Assembly line balancing using genetic algorithms. *Journal of Intelligent Manufacturing*, 11, 295-310.
- Safizadeh, M. H., Thornton, B. M., 1984. Optimization in Simulation Experiments Using Response Surface Methodology. *Computers & Industrial Engineering*, 8, 11-17.
- Salveson, M. E., 1955. The assembly line balancing problem. *Journal of Industrial Engineering*, 6, 18-25.
- Scholl, A., 1995. Balancing and sequencing of assembly lines. Darmstadt: Physica-Verlag.
- Simaria, A. S., 2006. Assembly line balancing - new perspectives and procedures. PhD Thesis, Universidade de Aveiro, Portugal.
- Siriwardene, N.R., Perera, B.J.C., 2006. Selection of genetic algorithm operators for urban drainage model parameter optimisation. *Mathematical and Computer Modeling*, 44, 415-429.
- Subbaraj, P., Rengaraj, R, Salivahanan, S., 2011. Enhancement of Self-adaptive real-coded genetic algorithm using Taguchi method for Economic dispatch problem. *Applied Soft Computing*, 11, 83-92.
- Vilarinho, P. M., Simaria, A. S., 2002. A two-stage heuristic method for balancing mixed-model assembly lines with parallel workstations. *International Journal of Production Research*, 40, 1405-1420.
- Yalcinkaya, O., Bayhan, G. M., 2009. Modeling and optimization of average travel time for a metro line by simulation and response surface methodology. *European Journal of Operational Research*, 196, 225–233.

- Yang, C. D., Luo, C.C., Liu, S.J., Chang, Y. H., 2005. Applications of Genetic-Taguchi Algorithm in Flight Control Designs. *Journal of Aerospace Engineering*, 18, 232, 241.
- Zaman, T., Paul, S. K., Azeem, A., 2012. Sustainable operator assignment in an assembly line using genetic algorithm. *International Journal of Production Research*, 50, 5077-5084.

Appendices

Obtained alternative solutions/task sequences (or chromosomes) for test problem 8 (25 tasks - 3 models) are given in following tables.

Accepted Manuscript

Tables and Figures List

Table 1. An overview of approaches in the literature on parameter optimisation of GAs.

Table 2. Levels and values of genetic algorithm parameters G_N , P_S , C_R , and M_R .

Table 3. Design of experiments matrix showing coded values and observed responses for 100 runs.

Table 4. Results of parameter optimisation.

Table 5. Test problems (Akpınar and Bayhan, 2011; Scholl, 1995; Vilarinho and Simaria, 2002).

Table 6. Computational results (station numbers) for test problem 8.

Table 7. Factor levels and values of genetic algorithm parameters for Taguchi design.

Table 8. Taguchi design and simulation results for average FV.

Table 9. Signal to noise (S/N) ratios for smaller is better criteria.

Table 10. Factor levels and values of genetic algorithm parameters for 2^k factorial design.

Table 11. 2^k factorial design and simulation results for average FV.

Table 12. Comparison of obtained results (station numbers) for test problems.

Table A.1 Alternative Solution-1.

Table A.2 Alternative Solution-2.

Table A.3 Alternative Solution-3.

Table A.4 Alternative Solution-4.

Table A.5 Alternative Solution-5.

Table A.6 Alternative Solution-6.

Figure 1. Flow chart of proposed genetic algorithm.

Figure 2. Optimisation results for G_N , P_S , C_R , and M_R .

Figure 3. Precedence diagram of test problem 8 (Simaria, 2006).

Figure 4. Optimisation results for the G_N , P_S , C_R and M_R for 2^k factorial design.

Table 1. An overview of approaches in the literature on parameter optimisation of GAs.

Publications	Method	Problem
Pongcharoen et al. (2002)	Factorial experiment, Regression analysis	Scheduling of complex products with multiple resource constraints
Lobo and Goldberg (2004)	Parameter-less genetic algorithm	Network expansion problem
Costa et al. (2005)	2^{k-1} fractional factorial design	Batch cooling crystallization optimization
Siriwardene and Perera (2006)	Proportionate selection, linear ranking selection	Urban drainage model parameter optimization
Yang et al. (2005)	Taguchi	Flight control design
Chang (2011)	Taguchi	Obtaining the steady state output voltage of proton exchange membrane fuel cell (PEMFC)
Fernandez-Prieto et al. (2011)	Fuzzy adaptive genetic algorithms – FAGAs	Computer networks under traffic loads
Subbaraj et al. (2011)	Taguchi self-adaptive real-coded genetic algorithm (TSARGA)	Economic dispatch problem

Table 2. Levels and values of genetic algorithm parameters G_N , P_S , C_R and M_R .

Parameter	Symbol	Level				
		-2	-1	0	1	2
Generation number	G_N	50	300	550	800	1050
Population size	P_S	20	40	60	80	100
Crossover rate	C_R	0.1	0.3	0.5	0.7	0.9
Mutation rate	M_R	0.05	0.1	0.15	0.2	0.25

Table 3. Design of experiments matrix showing coded values and observed responses for 100 runs.

Experiment Number	Run Order	Coded Value				Response Average FV
		G_N	P_S	C_R	M_R	
1	1	-1	-1	-1	-1	11.6584
2	2	1	-1	-1	-1	10.7626
3	3	-1	1	-1	-1	10.4247
4	4	1	1	-1	-1	9.9721
5	5	-1	-1	1	-1	10.7962
6	6	1	-1	1	-1	11.2854
7	7	-1	1	1	-1	10.0964
8	8	1	1	1	-1	10.4255
9	9	-1	-1	-1	1	10.8994
10	10	1	-1	-1	1	11.2969
11	11	-1	1	-1	1	11.0452
12	12	1	1	-1	1	9.6723
13	13	-1	-1	1	1	11.5418
14	14	1	-1	1	1	10.9198
15	15	-1	1	1	1	10.4896
16	16	1	1	1	1	9.4210
17	17	-2	0	0	0	11.2915
18	18	2	0	0	0	10.0103
19	19	0	-2	0	0	11.5502
20	20	0	2	0	0	10.1542
21	21	0	0	-2	0	10.9685
22	22	0	0	2	0	9.8641
23	23	0	0	0	-2	10.5057

24	24	0	0	0	2	9.4812
25	25	0	0	0	0	10.0597
26	26	0	0	0	0	10.0849
27	27	0	0	0	0	10.1023
28	28	0	0	0	0	10.1178
29	29	0	0	0	0	10.0195
30	30	0	0	0	0	10.0634
31	31	0	0	0	0	10.1041

Table 4. Results of parameter optimisation.

Symbol	Coded value	Uncoded value	Rounded value
G_N	1.9596	1039.9	1040
P_S	1.6364	92.73	93
C_R	0.2222	0.54	0.50
M_R	2	0.25	0.25

Table 5. Test problems (Akpınar and Bayhan, 2011; Scholl, 1995; Vilarinho and Simaria, 2002).

	Problem no	N	M	C	Precedence relations	Problem no	N	M	C	Precedence relations	
Small-size	1	8	2	10	Bowman	Medium-Size	11	30	2	10	Sawyer
	2	8	3	10	Bowman		12	30	3	10	Sawyer
	3	11	2	10	Gokcen and Erel (1998)		13	32	2	10	Lutz 1
	4	11	3	10	Gokcen and Erel (1998)		14	32	3	10	Lutz 1
Medium-Size	5	21	2	10	Mitchel	Large Size	15	35	2	10	Gunther
	6	21	3	10	Mitchel		16	35	3	10	Gunther
	7	25	2	10	Vilarinho and Simaria		17	45	2	10	Kilbridge & Wester
	8	25	3	10	Vilarinho and Simaria		18	45	3	10	Kilbridge & Wester
	9	28	2	10	Heskiakof		19	70	2	10	Tonge
	10	28	3	10	Heskiakof		20	70	3	10	Tonge

Table 6. Computational results (station numbers) for test problem 8.

#	Problem	LB_{pmix}	Kilbridge & Wester	Moodie & Young (Phase I)	RPWT	Pure GA	V&S (SA)	A&B (hGA)	RSM-GA
8	Vilarinho and Simaria (2002)	14	15	15	15	15	15	14	14

Table 7. Factor levels and values of genetic algorithm parameters for Taguchi design.

Parameter	Symbol	Level				
		1	2	3	4	5
Generation number	G_N	50	300	550	800	1050
Population size	P_S	20	40	60	80	100
Crossover rate	C_R	0.1	0.3	0.5	0.7	0.9
Mutation rate	M_R	0.05	0.1	0.15	0.2	0.25

Table 8. Taguchi design and simulation results for average FV.

Experiment Number	Run Order	Coded Value				Response
		G_N	P_S	C_R	M_R	Average FV
1	1	1	1	1	1	12.8754
2	2	1	2	2	2	10.2845
3	3	1	3	3	3	10.3232
4	4	1	4	4	4	10.0746
5	5	1	5	5	5	10.0551
6	6	2	1	2	3	12.5041
7	7	2	2	3	4	11.6579
8	8	2	3	4	5	9.7338
9	9	2	4	5	1	10.0792
10	10	2	5	1	2	9.2989
11	11	3	1	3	5	12.4054
12	12	3	2	4	1	10.9296
13	13	3	3	5	2	10.2612
14	14	3	4	1	3	11.1595
15	15	3	5	2	4	9.7173
16	16	4	1	4	2	12.5572
17	17	4	2	5	3	10.496
18	18	4	3	1	4	9.6257
19	19	4	4	2	5	10.9025
20	20	4	5	3	1	10.1299
21	21	5	1	5	4	12.4396
22	22	5	2	1	5	10.1224
23	23	5	3	2	1	10.9064
24	24	5	4	3	2	10.8023
25	25	5	5	4	3	10.1464

Table 9. Signal to noise (S/N) ratios for smaller is better criteria.

Level	G_N	P_S	C_R	M_R
1	-20.57	-21.98	-20.46	-20.78
2	-20.5	-20.57	-20.69	-20.5
3	-20.71	-20.14	-20.85	-20.74
4	-20.59	-20.5	-20.54	-20.54
5	-20.71	-19.88	-20.53	-20.51

Table 10. Factor levels and values of genetic algorithm parameters for 2^k factorial design.

Parameter	Symbol	Level	
		-1	1
Generation number	G_N	50	1050
Population size	P_S	20	100
Crossover rate	C_R	0.1	0.9
Mutation rate	M_R	0.05	0.25

Table 11. 2^k factorial design and simulation results for average FV.

Experiment Number	Run Order	Coded Value				Response
		G_N	P_S	C_R	M_R	Average FV
1	1	-1	-1	-1	-1	13.9733
2	2	1	-1	-1	-1	12.3377
3	3	-1	1	-1	-1	9.9266
4	4	1	1	-1	-1	10.9129
5	5	-1	-1	1	-1	13.4973
6	6	1	-1	1	-1	11.2922
7	7	-1	1	1	-1	9.7846
8	8	1	1	1	-1	11.8925
9	9	-1	-1	-1	1	12.9618
10	10	1	-1	-1	1	9.7963
11	11	-1	1	-1	1	9.4492
12	12	1	1	-1	1	9.6476
13	13	-1	-1	1	1	11.7615
14	14	1	-1	1	1	13.8808
15	15	-1	1	1	1	10.3125
16	25	1	1	1	1	10.2736

Table 12. Comparison of obtained results (station numbers) for test problems.

#	Problem	LB_{pmix}	Kilbridge & Wester	Moodie & Young (Phase I)	RPWT	Pure GA	V&S (SA)	A&B (hGA)	RSM-GA
P1	Bowman	4	4	4	4	4	4	4	4
P2	Bowman	6	9	9	9	8	8	8	8
P3	Gokcen and Erel (1998)	7	8	7	8	7	7	7	7
P4	Gokcen and Erel (1998)	6	7	7	7	7	7	7	7
P5	Mitchel	14	17	16	16	16	16	16	16
P6	Mitchel	13	16	15	17	15	15	15	15
P7	Vilarinho and Simaria (2002)	14	17	17	18	16	16	16	16
P8	Vilarinho and Simaria (2002)	14	15	15	15	15	15	14	14
P9	Heskiakof	19	21	21	21	21	21	20	20
P10	Heskiakof	18	21	21	21	20	20	20	20
P11	Sawyer	15	18	18	19	16	16	16	16
P12	Sawyer	17	21	21	20	19	19	19	19
P13	Lutz 1	16	19	19	19	19	19	19	19
P14	Lutz 1	17	21	20	20	19	19	19	19
P15	Gunther	20	25	25	24	24	24	23	24
P16	Gunther	21	26	26	25	25	24	24	24
P17	Kilbridge & Wester	23	27	27	28	25	25	25	25
P18	Kilbridge & Wester	24	31	29	29	28	28	27	27
P19	Tonge	41	50	48	48	45	44	44	44
P20	Tonge	39	48	47	48	45	44	44	45

Table A.1 Alternative solution-1.

Task	1	3	7	2	4	6	5	8	9	14	11	10	13
Station	1-2	1-2	1-2	3	3	3	4-5	4-5	4-5	6	6	7	7
t_A	4.1	4.6	11.3	2.7	4.1	0	2.0	7.8	0	5.1	3.9	3.5	2.5
t_B	4.1	4.6	11.3	2.7	4.1	2.0	2.0	7.8	10.0	5.1	4.2	3.5	2.3
t_C	4.1	4.6	11.3	2.7	4.1	2.0	2.0	7.8	10.0	5.1	3.9	3.3	2.5
$t_{Average}$	4.1	4.6	11.3	2.7	4.1	1.3	2.0	7.8	6.7	5.1	4.1	3.5	2.4
Task	16	18	19	12	17	20	21	22	15	23	24	25	
Station	7	8	9	10	10	11	12	12	13	13	13	14	
t_A	3.5	8.5	9.9	1.0	6.8	7.2	4.8	3.8	3.5	2.9	3.5	7.8	
t_B	3.5	8.5	9.9	1.0	6.8	7.2	4.8	3.8	3.5	2.8	3.5	7.8	
t_C	3.4	9.6	9.9	1.0	6.8	7.2	4.8	3.9	3.5	2.6	3.5	7.8	
$t_{Average}$	3.5	8.7	9.9	1.0	6.8	7.2	4.8	3.8	3.5	2.8	3.5	7.8	

Table A.2 Alternative solution-2.

Task	1	3	7	5	2	4	8	6	9	10	11	13	14
Station	1-2	1-2	1-2	3	3	3	4-5	4-5	4-5	6	6	6	7
t_A	4.1	4.6	11.3	2.0	2.7	4.1	7.8	0	0	3.5	3.9	2.5	5.1
t_B	4.1	4.6	11.3	2.0	2.7	4.1	7.8	2.0	10.0	3.5	4.2	2.3	5.1
t_C	4.1	4.6	11.3	2.0	2.7	4.1	7.8	2.0	10.0	3.3	3.9	2.5	5.1
$t_{Average}$	4.1	4.6	11.3	2.0	2.7	4.1	7.8	1.3	6.7	3.5	4.1	2.4	5.1
Task	16	18	17	20	21	22	12	19	15	23	24	25	
Station	7	8	9	10	11	11	11	12	13	13	13	14	
t_A	3.5	8.5	6.8	7.2	4.8	3.8	1.0	9.9	3.5	2.9	3.5	7.8	
t_B	3.5	8.5	6.8	7.2	4.8	3.8	1.0	9.9	3.5	2.8	3.5	7.8	
t_C	3.4	9.6	6.8	7.2	4.8	3.9	1.0	9.9	3.5	2.6	3.5	7.8	
$t_{Average}$	3.5	8.7	6.8	7.2	4.8	3.8	1.0	9.9	3.5	2.8	3.5	7.8	

Table A.3 Alternative solution-3.

Task	1	3	7	2	4	5	6	8	10	11	13	9	14
Station	1-2	1-2	1-2	3	3	3	4	4	5-6	5-6	5-6	7	7
t_A	4.1	4.6	11.3	2.7	4.1	2.0	0	7.8	3.5	3.9	2.5	0	5.1
t_B	4.1	4.6	11.3	2.7	4.1	2.0	2.0	7.8	3.5	4.2	2.3	10.0	5.1
t_C	4.1	4.6	11.3	2.7	4.1	2.0	2.0	7.8	3.3	3.9	2.5	10.0	5.1
$t_{Average}$	4.1	4.6	11.3	2.7	4.1	2.0	1.3	7.8	3.5	4.1	2.4	6.7	5.1
Task	12	16	17	21	22	18	19	20	15	23	24	25	
Station	7	7	8	9	9	10	11	12	13	13	13	14	
t_A	1.0	3.5	6.8	4.8	3.8	8.5	9.9	7.2	3.5	2.9	3.5	7.8	
t_B	1.0	3.5	6.8	4.8	3.8	8.5	9.9	7.2	3.5	2.8	3.5	7.8	
t_C	1.0	3.4	6.8	4.8	3.9	9.6	9.9	7.2	3.5	2.6	3.5	7.8	
$t_{Average}$	1.0	3.5	6.8	4.8	3.8	8.7	9.9	7.2	3.5	2.8	3.5	7.8	

Table A.4 Alternative solution-4.

Task	1	3	7	2	4	5	9	8	6	12	14	10	11
Station	1-2	1-2	1-2	3-4	3-4	3-4	3-4	5	5	6	6	6	7
t_A	4.1	4.6	11.3	2.7	4.1	2.0	0	7.8	0	1.0	5.1	3.5	3.9
t_B	4.1	4.6	11.3	2.7	4.1	2.0	10.0	7.8	2.0	1.0	5.1	3.5	4.2
t_C	4.1	4.6	11.3	2.7	4.1	2.0	10.0	7.8	2.0	1.0	5.1	3.3	3.9
$t_{Average}$	4.1	4.6	11.3	2.7	4.1	2.0	6.7	7.8	1.3	1.0	5.1	3.5	4.1
Task	13	16	18	20	17	21	22	19	15	23	24	25	
Station	7	7	8	9	10	11	11	12	13	13	13	14	
t_A	2.5	3.5	8.5	7.2	6.8	4.8	3.8	9.9	3.5	2.9	3.5	7.8	
t_B	2.3	3.5	8.5	7.2	6.8	4.8	3.8	9.9	3.5	2.8	3.5	7.8	
t_C	2.5	3.4	9.6	7.2	6.8	4.8	3.9	9.9	3.5	2.6	3.5	7.8	
$t_{Average}$	2.4	3.5	8.7	7.2	6.8	4.8	3.8	9.9	3.5	2.8	3.5	7.8	

Table A.5 Alternative solution-5.

Task	1	3	7	2	4	5	8	6	10	11	13	9	16
Station	1-2	1-2	1-2	3	3	3	4	4	5-6	5-6	5-6	7	7
t_A	4.1	4.6	11.3	2.7	4.1	2.0	7.8	0	3.5	3.9	2.5	0	3.5
t_B	4.1	4.6	11.3	2.7	4.1	2.0	7.8	2.0	3.5	4.2	2.3	10.0	3.5
t_C	4.1	4.6	11.3	2.7	4.1	2.0	7.8	2.0	3.3	3.9	2.5	10.0	3.4
$t_{Average}$	4.1	4.6	11.3	2.7	4.1	2.0	7.8	1.3	3.5	4.1	2.4	6.7	3.5
Task	14	17	18	19	20	21	22	12	15	23	24	25	
Station	7	8	9	10	11	12	12	12	13	13	13	14	
t_A	5.1	6.8	8.5	9.9	7.2	4.8	3.8	1.0	3.5	2.9	3.5	7.8	
t_B	5.1	6.8	8.5	9.9	7.2	4.8	3.8	1.0	3.5	2.8	3.5	7.8	
t_C	5.1	6.8	9.6	9.9	7.2	4.8	3.9	1.0	3.5	2.6	3.5	7.8	
$t_{Average}$	5.1	6.8	8.7	9.9	7.2	4.8	3.8	1.0	3.5	2.8	3.5	7.8	

Table A.6 Alternative solution-6.

Task	1	3	7	2	5	4	9	8	6	10	14	11	13
Station	1-2	1-2	1-2	3-4	3-4	3-4	3-4	5	5	6	6	7	7
t_A	4.1	4.6	11.3	2.7	2.0	4.1	0	7.8	0	3.5	5.1	3.9	2.5
t_B	4.1	4.6	11.3	2.7	2.0	4.1	10.0	7.8	2.0	3.5	5.1	4.2	2.3
t_C	4.1	4.6	11.3	2.7	2.0	4.1	10.0	7.8	2.0	3.3	5.1	3.9	2.5
$t_{Average}$	4.1	4.6	11.3	2.7	2.0	4.1	6.7	7.8	1.3	3.5	5.1	4.1	2.4
Task	16	17	18	20	19	12	21	22	15	23	24	25	
Station	7	8	9	10	11	12	12	12	13	13	13	14	
t_A	3.5	6.8	8.5	7.2	9.9	1.0	4.8	3.8	3.5	2.9	3.5	7.8	
t_B	3.5	6.8	8.5	7.2	9.9	1.0	4.8	3.8	3.5	2.8	3.5	7.8	
t_C	3.4	6.8	9.6	7.2	9.9	1.0	4.8	3.9	3.5	2.6	3.5	7.8	
$t_{Average}$	3.5	6.8	8.7	7.2	9.9	1.0	4.8	3.8	3.5	2.8	3.5	7.8	

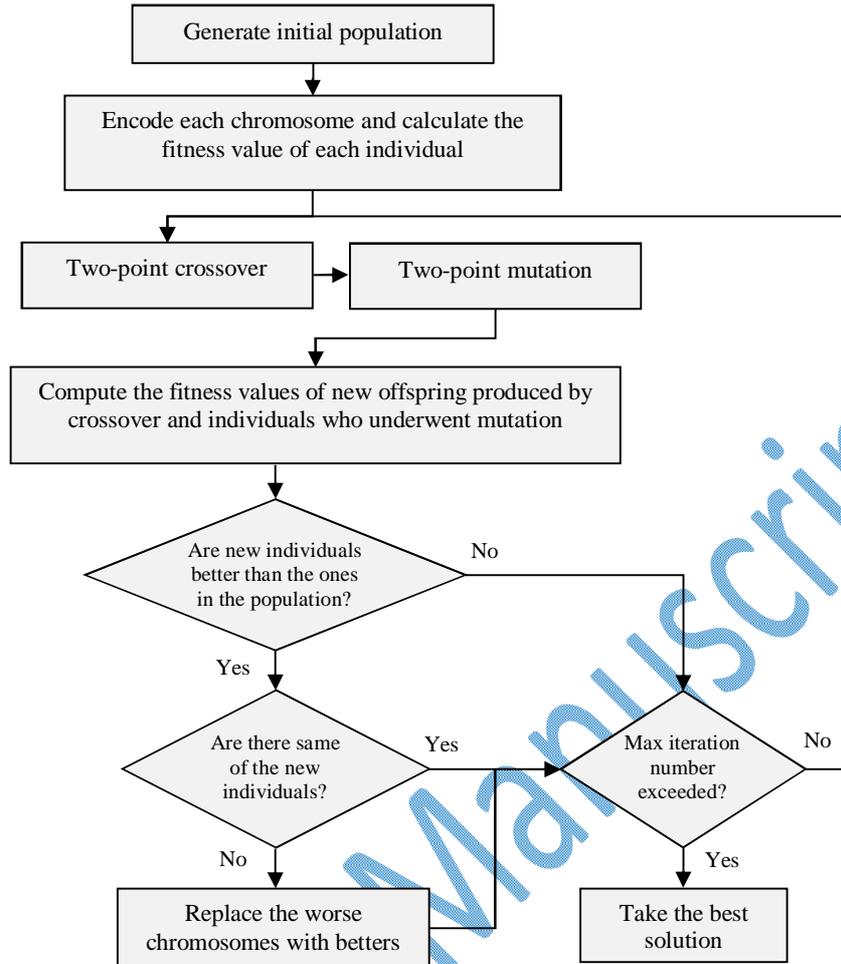


Figure 1. Flow chart of proposed genetic algorithm.

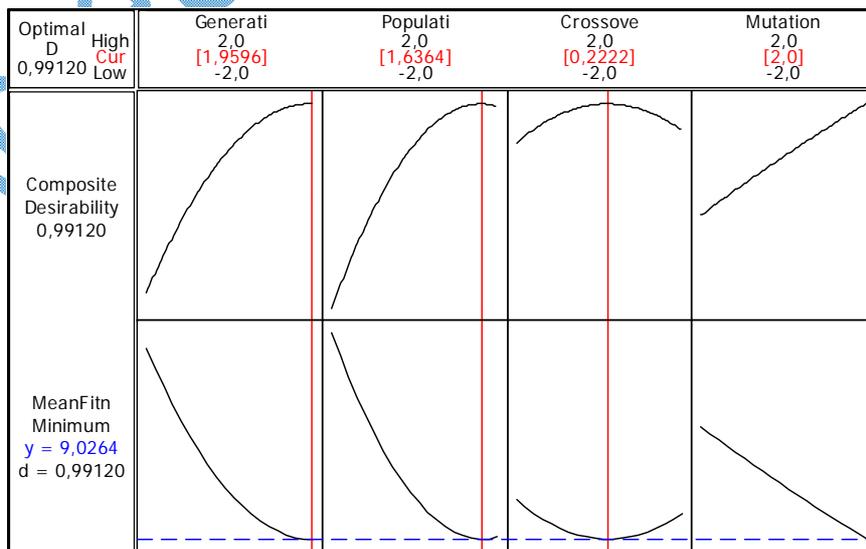


Figure 2. Optimization results for G_N , P_S , C_R and M_R .

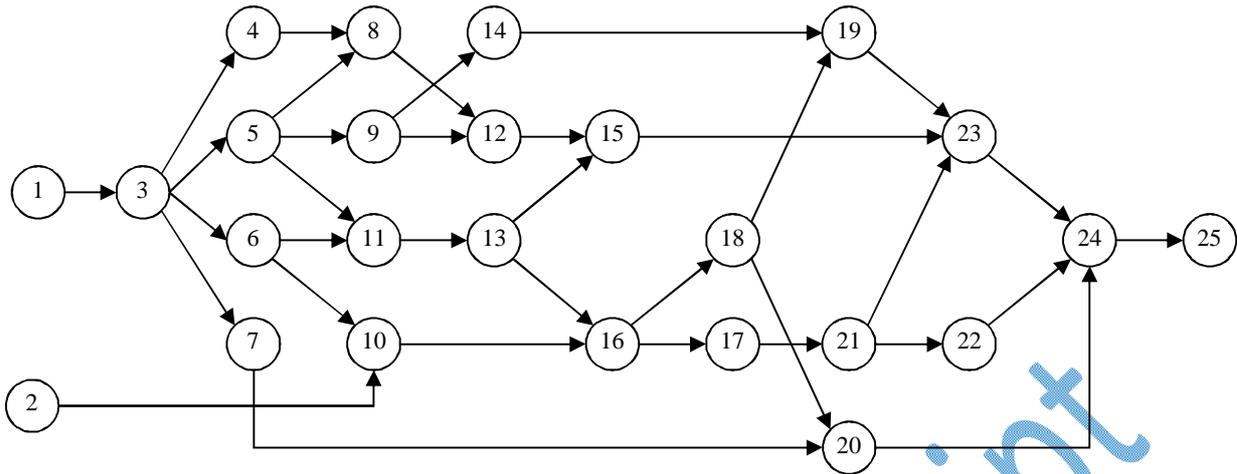


Figure 3. Precedence diagram of test problem 8 (Simaria, 2006).

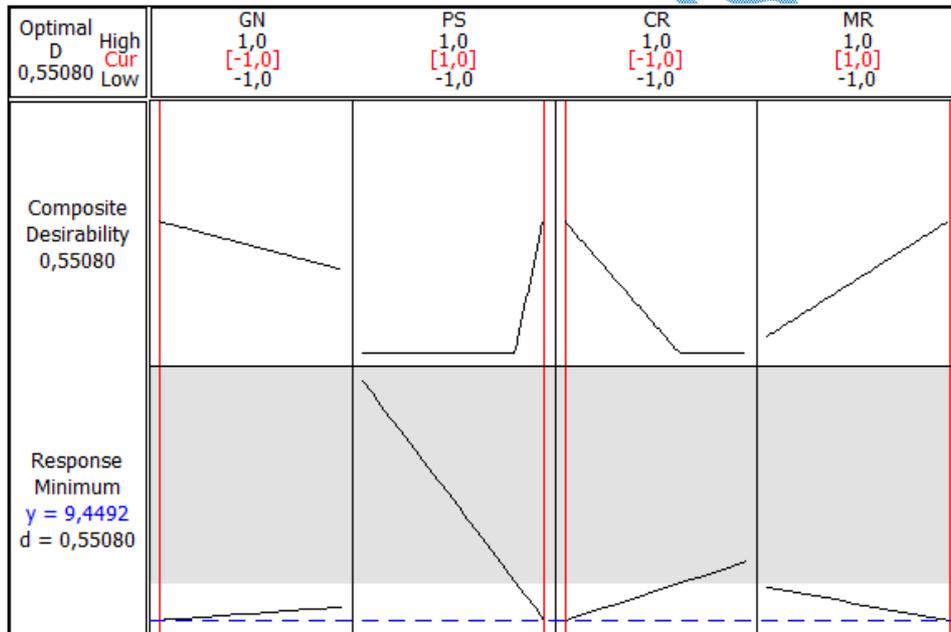


Figure 4. Optimization results for the G_N, P_S, C_R and M_R for 2^k factorial design.