ABSTRACT

Just-In-Time (JIT) as a lean manufacturing approach plays a significant role in minimising costs and performances of products and services supplied to the global marketplace. However, there are many potential risks that cause significant disruptions to all supply chain members. This study proposes a genetic approach for optimising a novel mathematical model for simultaneously minimising the total cost of a final product and the potential risks related to these benefits. Specifically, it demonstrates the effectiveness of a genetic algorithm in optimising the JIT model developed in our previous paper. Genetic operators adopted to improve the genetic search algorithm are introduced and discussed. Experiments are carried out to evaluate the performance of the proposed algorithm using a simplified example. Comparison of four selection methods is done to define the best method that can be used in the proposed GA. The findings demonstrate the superiority of the proposed approach in the JIT system with focus on simultaneous cost-risk reduction.

Keywords: Just-In-Time (JIT), Production System, Cost-Risk Reduction, Model, Optimisation, Genetic Algorithm (GA)

1. INTRODUCTION

Optimisation problems arise in case discrete choices must be made and solving them amounts to find an optimal solution among a large number of alternatives. Recently, researchers have proposed many approaches that can be used for obtaining the optimum solutions within a reasonable amount of time (Tasan and Tunali, 2008). Recently, many meta-heuristics optimisation approaches have been applied in the supply chain management field to solve engineering optimisation issues (Wang and Wang, 2008). These approaches are associated with transportation/distribution networks consider the location of the organisation, design of the network configuration and customer satisfaction by maximising the total product cost (Paksoy and Chang, 2010). Optimisation is the process of adapting the inputs of a device and mathematical process to find the minimum or maximum of the output. Optimisation is a main technique that can be used for addressing complex problems. The key purpose of optimisation is to find the global optima (maximum or minimum) of a formulated objective function to a problem. Optimisation problems are used for acquiring good component parameters to be set into activities by humans or machines (Malhotra et al., 2011).

Numerous industrial engineering design problems are very complex and hence intractable for conventional optimisation techniques. Evolutionary Algorithms (EAs) are population-based meta-heuristic optimisation
algorithms that refine a set of solution iteratively using biology-inspired mechanisms and theory of the survival fittest (Malhotra et al., 2011). The evolution process initiates from a population of generated individuals and occurs in generations (Kannaiah et al., 2011).

Globally, many organisations have used a JIT approach in their processes to achieve their objectives. However, some have ignored certain significant risks which arise from its implementation such as natural disasters. Businesses are exposed to numerous risks, particularly those originating from the supply chain. These risks potentially influence their processes by disrupting all supply chain members involved (El Dabee et al., 2013b; 2013c). In this context, risk is defined as the combination of possible consequences and associated uncertainties (Aven and Vinnem, 2007). Generally, supply chain risk is an event exposure that causes disruption affecting the efficient processes of the whole supply chain (Ghadge et al., 2012).

The main objective of this study is to develop an optimisation Approach (GA) for the proposed JIT model published in (El Dabee et al., 2013a). The novelty of this study is the simultaneous cost-risk reduction of the final product in the production system under a JIT environment by using the GA approach for finding the optimum solution of the proposed model.

In the remainder of this study, Subsection 1.1 reviews the literature on JIT approach using optimisation techniques and Subsection 1.2% an overview of the JIT model reported in our previous study in (El Dabee et al., 2013a); section 2 presents the materials and methods adopted in this study includes the proposed GA parameters for optimising the JIT model and implementing the proposed GA in a simplified example to find the optimal or near-optimal solution of cost-risk reduction in JIT systems; section 3 discusses and analyses the obtained results to illustrates the effectiveness of the proposed GA; while section 4 summarises the paper.

1.1. An Overview of JIT Optimisation Using GA

In today’s increasingly competitive environment, the adoption of efficient approaches plays a decisive role in successful performance of organisations (Cai et al., 2009). This can be achieved by the continuous improvement and optimisation of processes and operations, cost reduction of services and products and an increased outputs capacity with satisfactory product quality and production rates (El Dabee and Hokoma, 2012; Hokoma et al., 2010). JIT as one of the most significant approaches can be used for eliminating all waste and non-value added activities within organisations, in order to minimise products cost (Krajewski et al., 2013).

Recently, GA as an evolutionary or met-heuristic algorithm has received a substantial attention as robust optimisation techniques to find the optimal or near-optimal solution within a reasonable amount of time (Geem et al., 2005; Sivanandam and Deepa, 2008; Yussof et al., 2011). GA is a subclass of Evolutionary Algorithms (EAs) where the elements of the search space are strings or arrays of other elementary types (Rawal and Inamdar, 2014). Based on their simplicity, ease of operation, minimal requirements and parallel and global perspective, GA has been widely applied in a variety of problems (Sivanandam and Deepa, 2008). GA can optimise with discrete, continuous and mixed parameters cope with a large number of parameters; and do not require derivative information (Marian, 2003; Park et al., 2011).

A JIT distribution model for three-level supply chain network was developed to minimise the product cost, the sum of backorders and product surpluses during all stages in JIT distribution systems. The model considers distribution lead time and capability restrictions in multiple periods, products and channel networks. A HGA was designed to solve real large-size problems of the proposed model. The results obtained from some small-size test problems were compared with the results obtained by LINGO optimisation software (Farahani and Elahipanah, 2008). GA and fuzzy set approach was applied to generate JIT scheduling to optimise the fabric-cutting process in the apparel industry. Data was collected from fabric-cutting department of Hong Kong-owned apparel manufacturing company to validate the proposed optimisation method. The findings illustrated the effectiveness of this approach, which improve the performance and minimise the production cost within the organisation (Wong et al., 2006).

Ghasimi et al. (2014) introduced a novel mathematical model for three-echelon Defective Goods Supply Chain Network (DGSCN) was built to ascertain the Economic Production Quantity (EPQ) and lead time, additionally reducing all production, distribution and holding cost using JIT logistics. GA and Cplex solver were used to optimise their defective goods supply chain network model. The model outputs were compared to define its performance and find the proper Length for Each Cycle (ALOEC). Jianhua and Xianfeng (2010)
proposed a Hybridised Genetic Algorithm (HGA) by merging GA with the Greedy Algorithm (GrA) to solve the Agile Supply Chain Scheduling (ASCS) problem in order to reduce the inventory and transportation cost in the supply chain. The authors showed that practical production and transportation schedules ensure the implementation of JIT while reducing the costs of operation in the supply chain.

A Multi-Objective Genetic Algorithm (MOGA) approach in JIT sequencing problem environment was proposed. MOGA was used to simultaneously minimise the number of required setups and the production rate variation. Extensive computational experiments were applied on the proposed algorithm to demonstrate its effectiveness. The results showed that this approach has a significant impact on the quality improvement compared with Total Enumeration (TE) approach in a short time (Mansouri, 2005).

According to the aforementioned literature related to a GA optimisation, there is no research to date addresses the simultaneous cost-risk reduction in the supply chain, particularly, JIT systems.

1.2. The Proposed Model

This Subsection presents the main optimisation function of the proposed model developed for simultaneously minimising the cost of the final product and the risks effect in a production system under JIT environment in (El Dabee et al., 2013a). This model incorporates four main parties, which are supplier, manufacturer, distribution centre and end user. It is assumed that a distribution network consists of multiple external suppliers who supply raw materials or components/subassemblies to the production system to produce the final product. This assumption is based on the pricing variations for the same product in different markets. The materials are transported from different manufacturers to the production system, which in turn produce the final product for sale to wholesale or retail outlets. Also, the raw materials are supplied instantaneously to the production system to meet JIT requirements. The products are delivered to the end customers such as wholesalers or retailers with no holding capacity to store the products.

Many risks may result from unforeseen disruptions such as natural and man-made disasters and economic crises affecting external suppliers. All of them have a significant impact on the production facility and the entire supply chain as well. To avoid the impact of these risks, it is assumed that during a time of disruption, the production system can procure its raw materials or components/subassemblies required to produce the final product from a local backup supplier at a higher price but with low risk and in a short lead time (El Dabee et al., 2013b).

The model also considers scenarios in case orders for raw materials are shipped by both suppliers using different transportation modes: Waterways, railways, roadways and the airways (Murphy and Wood, 2011). The appropriate transportation mode can be selected depending on some key criteria such as the shortest time and the transportation cost required for carrying raw materials to the production system. The model is coded to ascertain the total cost of producing the final product within production systems.

The following notations are used in the proposed model:

- \( C_F \): Total cost required to produce one product in Monetary Unit (MU)
- \( C_{PO} \): Final product cost excluding the risk cost in Monetary Unit (MU)
- \( C_{RM} \): Raw material cost required for producing one product (MU)
- \( C_{OR} \): Ordering cost of raw Materials (MU)
- \( C_{HR} \): Holding cost of raw materials within the production system stores (MU)
- \( C_{CR} \): Risk cost arising from disruption occurrence (MU)
- \( C_{TR} \): Transportation cost for delivering raw materials to the production system (MU)
- \( C_{UR} \): Purchasing cost of raw materials required to produce the product (MU)
- \( C_{US} \): Utilities cost of the final product (MU)
- \( TP_i \): Transfer price required for procuring raw material \( i \) from an external supplier \( j \) (MU)
- \( S \): Origin of ordered raw materials
- \( v \): Destination of required raw materials
- \( m_i \): Transportation mode for transporting raw material \( i \) to its customer
- \( N_T \): Number of transportation modes used for shipping raw materials to the production system
- \( S_{Ej} \): Raw material external supplier \( j \)
- \( S_{EBs} \): Raw material local backup supplier \( s \)
- \( IF \): Indicator function for duty with a value 1 or 0.1 if the supplier and the production facility are in the same country and 0 otherwise
• **LH**: Likelihood of occurrence for risk in the supply chain  
• **I**: Impact of risk occurrence in the supply chain  
• **N_O**: Number of operations required for producing one product (unit)  
• **N_P**: Number of parts required to produce one product  
• **N_SLB**: Number of local suppliers used for supplying raw materials to the production system (unit)  
• **C_UO**: Ordering cost of raw materials for the final product (MU)  
• **C_UH**: Holding cost of raw materials of each final product in the production system warehouse (MU/day)  
• **C_UM**: Unit cost of the raw material at the beginning of each cycle (MU)  
• **%d_RM**: Daily demand percentage of raw materials required to meet customer satisfaction (unit)  
• **h**: Operation time required to produce a product (hr)  
• **C_L**: Labour cost rate per time in one operation (MU/hr)  
• **SF**: Storage factor for keeping raw materials in the warehouse  
• **OF**: Ordering factor for procuring each order from the supplier  
• **%Util**: Utilities cost percentage of the final product  
• **t_p**: Raw material cost percentage incurred for procuring raw material from an external supplier  
• **LT**: Lead-time taken between placing and receiving the placed order (day)  
• **T_{SLB, v, m}**: Tensor for transportation cost per critical measurement to transport raw materials from its origin to its destination using transportation mode (MU)  
• **%V**: Volume percentage value required for transporting raw materials to their customer  
• **D_j**: Duty rate (%) per price of raw material supplied by external supplier  
• **%TRS**: Total risk score percentage value

The purpose of this model is to optimise the total costs ($C_T$), including $C_O$, $C_H$, $C_R$, $C_P$, $C_U$, $C_T$, $C_W$, $C_U$, and $C_R$ using the GA technique. The total cost of this product can be calculated as Equation 1 and 2:

\[ C_T = C_P + C_R \]  
\[ C_P = C_O + C_H + C_U + C_D + TP + C_W + C_U \]

$C_T$ can be calculated in the case of using the external supplier for procuring raw materials as follows Equation 3:

\[ C_T = \sum_{j=1}^{N_{SE}} C_{O_{ij}} \times OF + \sum_{j=1}^{N_{SE}} \sum_{i=1}^{N_P} (C_{RM_{ij}}) \times \%d_{RM} \times \left( LT_{ij} + SF \right) + \sum_{j=1}^{N_{SE}} \sum_{i=1}^{N_P} \sum_{k=1}^{N_{SLB}} \sum_{v=1}^{N_{RM}} (C_{U_{RM_{ij,k,v}}}) \times \%V_{ij,k,v} \times D_j + \sum_{j=1}^{N_{SE}} \sum_{i=1}^{N_P} \sum_{k=1}^{N_{SLB}} \sum_{v=1}^{N_{RM}} T_{SLB_{ij,k,v}} \times \%V_{ij,k,v} \times \%Util \times C_{RM_{ij,k,v}} + \sum_{i=1}^{N_P} C_{U_{ij}} + \sum_{i=1}^{N_P} LH_j \times I_s \times \%Util \times C_{RM_{ij}} \times \%TRS \]

Also, when raw materials are supplied by the local backup supplier, $C_T$ can be found as Equation 4:

\[ C_T = \sum_{i=1}^{N_{SLB}} C_{U_{i}} \times OF + \sum_{i=1}^{N_{SLB}} \sum_{k=1}^{N_{RM}} (C_{U_{RM_{i,k}}}) \times \%d_{RM} \times \left( LT_i + SF \right) + \sum_{i=1}^{N_{SLB}} \sum_{k=1}^{N_{RM}} \sum_{v=1}^{N_{RM}} T_{SLB_{i,k,v}} \times \%V_{i,k,v} \times \%Util \times C_{RM_{i,k}} + \sum_{i=1}^{N_P} C_{U_{i}} + \sum_{i=1}^{N_P} LH_i \times I_s \times \%Util \times C_{RM_{i}} \times \%TRS \]

2. MATERIALS AND METHODS

GA is a modern heuristic algorithm, which is widely adopted by researchers in solving complex and large-scale combinatorial optimisation problems (Littidej and Suttayamully, 2009; Radhakrishnan et al., 2009; Ali et al., 2010). GA is an adaptive heuristic based on ideas of natural selection and genetics and one of the most popular approaches of evolutionary algorithms (Gupta and Ghafir, 2012). GA is the key paradigm of evolutionary computation, which is inspired by Darwin’s theory of the survival of the fittest (Scavino et al., 2009; Zhu and Kwong, 2010). GA retains a population of different solutions that allows them to mate, produce offspring, mutate and fight for survival. The principle of survival of the fittest ensures the populations derive towards optimisation (Wang and Wang, 2008).

The proposed GA was coded using Java language and run on a PC Intel® Core™ i5-3210M processor working at speed 2.5 GHz and RAM of 4GB. This study uses GAs to study the simultaneous cost-risk reduction under the JIT approach to find the optimal solution of the developed model used for this purpose, which has been tested with a simplified example. The
key components of GA are: The representation of the solution, creation of initial population, evaluation of solutions, the development of genetic operators (crossover and mutation) and the termination creation (Go et al., 2010; Goren et al., 2010). Also, Mahmudy et al. (2012) summarise the main stages of GA as the key parameters setting, initialisation, reproduction and selection stages. All these stages will be described briefly as follows.

2.1. Parameters Setting

Setting GA parameters is the first stage that includes the population size (N), crossover rate (Pc), mutation rate (Pm), maximum Generations (Gmax) and the number of generations after which the GA will stop if no further improvement for the last best solution (generations after which the GA will stop if no further improvement for the last best solution (GStop)) summarise the main stages of GA as the key parameters setting, initialisation, reproduction and selection stages. All these stages will be described briefly as follows.

2.2. Initial Population Generation

GA starts with generation of a number of solutions (chromosomes), called a population. The population is usually randomly initialized (Al-Taharwa et al., 2008; Kumar et al., 2012). This stage is a key issue in GA approach based on their significant impacts on the subsequent stages. Selecting a good initial population with an appropriate population size can greatly improve the GA efficiency (Torabi et al., 2006). Each chromosome/individual is constituted by genes which are generated with random values. Chromosomes evolve through successive iterations, called generations.

2.2.1. Chromosome Representation

The first and one of the most significant steps in applying GA to a particular problem is to convert solutions (chromosomes) of a JIT cost-risk reduction problem into a string type structure called chromosome. Each chromosome represents one potential solution. The chromosome is a string of symbols that can be coded in different forms such as binary, integer and real numbers (Gen and Cheng 2000). The chromosome refers to the random population of encoded candidate solutions with which the Genetic algorithms initiate with (Radhakrishnan et al., 2009). This study uses a real number representation, called Real-Coded Genetic Algorithms (RCGA). RCGA considers each chromosome as a vector of real numbers. For a problem of n variables, the structural representation of a chromosome, X is: \( X = (X_1 + X_2 + X_3 + \ldots \ldots + X_n) \) (Gen and Cheng, 2000). Here, the chromosome of 4 genomes (X1, X2, X3 and X4) represents the main decision variables with 34 genes. X1 presents the number of external suppliers (S1) used for supplying raw materials to the production system, which are 11 suppliers (genes) with 2 levels, if \( X_1 = 1 \). S1 can supply raw materials and \( C_T \) is calculated using Equation 3 and if \( X_1 = 0 \), S1 has disruption and \( C_T \) is calculated using Equation 4; \( X_2 \) presents the customer demand (dp) for the final product per day with 1 demand (gene), which has 210 levels (\( d_p = 1, 2, \ldots, 210 \)); \( X_3 \) presents the critical transportation measurement (\( t_m \)) of raw materials shipped using transportation mode \( m \) with 11 suppliers (genes), which has 4 levels (\( m = 1, 2, 3, 4 \)); finally, \( X_4 \) presents the quantity of raw material (\( Q_m \)) ordered in each patch required to produce the final product per week using 11 suppliers (genes) with 7 levels (\( Q_m = 350, 700, 1050, 1400, 1750, 2100, 2450 \)). A random individual generated for the genetic operation is shown in Table 1.

By generating the individuals, the number of occurrences of the individual in the previous records is determined. Based on the four decision variables used, the number of solutions can be calculated as \( 2^{11} \times 210 \times 4^{11} \times 7^{11} = 3.57 \times 10^{12} \) Solutions.

2.2.2. Fitness Function

A fitness function is required to evaluate the quality of each chromosome. Generally, the fitness function is derived from the objective function and used in successive processes (Garg, 2010). The objective function of the proposed model is to minimise the total cost of the final product, which includes the product cost and the risk cost resulting from this reduction. To convert the objective function (minimum) illustrated in Equation 3 and 4 to a fitness objective (maximum), the fitness function is calculated as follows Equation 5:

\[
F = \frac{100}{(C_T + 1)} = \frac{100}{\left(\sum_{i=1}^{n}C_{p_i} + \sum_{i=1}^{n}C_{d_i}\right) + 1} \quad (5)
\]

The reason of adding the constant 1 to the equation is to prevent dividing the value by zero.

2.3. Reproduction

In this stage, genetic operators such as crossover and mutation are used to explore the search space, in order to create new chromosomes (offsprings). The number of new chromosomes (N) in the offspring pool is determined by the crossover rate (Pc) and mutation rate (Pm) parameters considering \( P_c + P_m = \text{constant} \).
Table 1. Chromosome construction

|   | X1 | X2 | X3 | X4 |
|---|----|----|----|----|
| 1 | 0  | 1  | 0  | 1  |
| 2 | 0  | 1  | 0  | 1  |
| 3 | 1  | 3  | 4  | 2  |
| 4 | 2  | 4  | 1  | 3  |
| 5 | 4  | 2  | 6  | 1  |
| 6 | 3  | 7  | 2  | 5  |
| 7 | 3  | 4  |    |    |

2.4. Crossover

The crossover operator, which is the basic operator of GA, mates two chromosomes to produce new chromosome (offspring). The key purpose of using crossover operator is to recombine the features of two parents selected randomly from the mating pool to produce a new chromosome (Abdelmaguid and Dessouky, 2006; Jalilzadeh et al., 2009). During the crossover, one or more genes will be exchanged between this pair of chromosomes. The crossover points are usually determined in a random manner. Crossover rate is another significant parameter in GA approach. A higher, but not too higher crossover rate permits more exploration of the solution space and minimise the opportunities of setting at a local optima (Mohd-Lair, 2008). This study uses three crossover methods: One-cut point crossover, flat-crossover and extended-intermediate-crossover. For each generation, one method is randomly selected to produce varied offspring.

2.4.1. Mutation

Mutation is a background operator which creates spontaneous random changes in numerous chromosomes. It is used to enhance the algorithm exploratory character and to avoid the GA from being trapped in local optima. Mutation operator helps the GA to avoid premature convergence and find the optimal solution in the search space based on the mutation rate \( P_m \) (Abdelmaguid and Dessouky, 2006; Leuveano et al., 2012). The main purpose of mutation operator is to produce little perturbations on chromosomes to maintain the diversity in the population.

The mutation rate also has a significant role in GA development. Therefore, defining the most appropriate crossover rate is necessary. To ensure the feasibility of mutated chromosomes, three methods of mutation are used: Simple random mutation, random exchange mutation and random insertion mutation. In each generation, one mutation is randomly selected.

2.4.2. Selection

The selection operator implements the idea of “the survival of the fittest”. It is the process of selecting chromosomes from the solution pool into a mating pool for generating offspring (Abdelmaguid and Dessouky, 2006). Selection operator describes the process of choosing a part of chromosomes from the initial population to keep their features in the next generation (Gen and Cheng, 1997). The main objective of selection strategy is to choose only the best chromosomes from the parents and offspring pools that can represent the solution space (Mohd-Lair, 2008). In this study, four selection methods (Roulette wheel, Binary tournament, Elist and Replacement selection methods) are tested and the results illustrate that the replacement selection method is the best method that can be used for the proposed GA. The replacement selection method proves that the best chromosome always passes to the next generation. To avoid premature convergence (maintain population diversity), the mechanism strategy is also applied to adjust crossover and mutation rates.

2.4.3. Simplified Example

The proposed mathematical model has been tested with a simple assembly process of an electric motor with hollow shaft. It uses multiple, identical operations to assemble 25 individual parts into the finished product. The proposed GA is used to find the optimal or near-optimal solution of the problem. It is assumed that a production system purchases raw materials in a fixed lot size from eleven different regular external suppliers. It is also assumed that in the case of one or more of external suppliers suffering disruption, the production system can procure its raw materials from seven local backup suppliers. The details of the simplified example are presented in (El Dabee et al., 2013b). Based on some experiments using different values of the GA basic parameters, the appropriate input parameters of the proposed GA are as shown in Table 2.

New chromosomes are produced by using crossover and mutation. Therefore, the crossover produces new chromosomes equal \( N \times P_c \) offspring and the mutation produces \( N \times P_m \) offspring for each generation. By using the crossover rate of 0.8, the number of chromosomes is \( 400 \times 0.8 = 320 \) offspring and the number of new chromosomes produced using the mutation rate of 0.2 is \( 400 \times 0.2 = 80 \) offspring in each generation.

Table 2. Input GA parameters

| Parameter                   | Rate  |
|-----------------------------|-------|
| Crossover rate \( (P_c) \)  | 0.8   |
| Mutation rate \( (P_m) \)   | 0.2   |
| Population size \( (N) \)   | 400.0 |
| Maximum generations \( (G_{max}) \) | 5000.0 |
| Generation stop \( (G_{max,Stop}) \) | 30000.0 |
3. RESULTS

The proposed GA tests the four selection methods (Roulette wheel, Binary tournament, Elitist and Replacement selection methods). A comparison between the four selection methods is carried out to determine which method is more suitable than the others by using the GA parameters illustrated in Table 2. The RCGA is run 10 times for each method to obtain the optimum solution for minimising the total cost of the final product and the risk effects in JIT systems in a short time. Table 3 illustrates the average values of the best iterations, computation time, customer demand, product cost, risk cost, total cost (maximum and average) and fitness function (maximum and average) for each selection method.

4. DISCUSSION

The computational results illustrated in Table 3 indicates that the parameters \( d_P \), \( C_{pt} \), \( C_R \), \( C_{T-max} \) and \( F.F_{max} \) for all selection methods have the same values. However, the other values, which are the number of iterations (iter.), computation time \( (T) \), \( C_{T-ave} \) and \( F.F_{ave} \) are different. These values show that the replacement selection method is the best methods compared with the other methods by achieving the higher average value of fitness values and minimum total cost.

Figure 1 depicts the comparison of the average of maximum generations of the ten runs between the four selection methods. It is clear that using the replacement method, the average value of maximum generations is higher than the others.

Fig. 2 shows the average and best costs of all chromosomes related to the number of GA iterations. The injection strategy is used to maintain the population diversity. Form Fig. 2, it is clear that the injection strategy has caused a fluctuation on the average values since new chromosomes that may have lower fitness values are injected into the population.

Using the best iteration of the ten runs for each selection method, Fig. 3 illustrates a comparison between the optimum solutions obtained by the four selection methods. It can be seen that the maximum generation is by using the replacement selection method.

Adopting the appropriate selection method (replacement method), the GA was run 10 times and from the obtained results, the optimum solution is achieved after 360 generations and the best chromosome is shown in Table 4. The findings of the proposed GA are listed in Table 5 including the number of iterations, the required time, the customer demand and the cost types.
Fig. 2. The average and best cost of the final product

Fig. 3. The best generation for each selection method
Table 3. The GA outputs of the four selection methods

| Method | Iter. | Time (S) | d_p | C_et | C_R | C_T-max | C_T-ave | F_F-max | F_F-ave |
|--------|-------|----------|-----|------|-----|---------|---------|---------|---------|
| RW     | 90    | 47.2     | 210 | 418.573 | 20.93 | 439.5013 | 446.8472 | 0.227014 | 0.223667 |
| BT     | 122   | 47.0     | 210 | 418.573 | 20.93 | 439.5013 | 446.8472 | 0.227014 | 0.223667 |
| ELIT   | 110   | 46.9     | 210 | 418.573 | 20.93 | 439.5013 | 442.1773 | 0.227014 | 0.225643 |
| REP    | 227   | 47.1     | 210 | 418.573 | 20.93 | 439.5013 | 440.1363 | 0.227014 | 0.226687 |

Table 4. The best chromosome using a replacement selection method

| X1 | X2 | X3 | X4 |
|----|----|----|----|
| 1  | 0  | 0  | 1  |
| 1  | 0  | 1  | 1  |
| 1  | 1  | 1  | 1  |
| 1  | 1  | 1  | 1  |
| 1  | 1  | 1  | 1  |
| 1  | 1  | 1  | 1  |
| 1  | 1  | 1  | 1  |
| 1  | 1  | 1  | 1  |

5. CONCLUSION

This study presented a genetic algorithm approach to solve the problem for the objective of a simultaneous cost-risk reduction in JIT systems. The proposed GA used the RCGA to search for an optimum solution in a short time. Experiments were carried out on the proposed GA using four selection methods to determine the best method that can be used. The findings demonstrated that a replacement selection method is more effective than the other methods in solving such issue by its higher average of fitness function and lower average value of the total final product cost.

Owing to there is no previous models compared with our model/results except for the previous paper from our research, our future work will focus on further experiments wherein different GA parameter settings are explored and their outcomes compared with other optimisation approaches.

6. ACKNOWLEDGMENT

The researchers would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

6.2. Author’s Contributions

All authors equally contributed in this work.

6.3. Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

7. REFERENCES

Abdelmaguid, T. and M. Dessouky, 2006. A genetic algorithm approach to the integrated inventory-distribution problem. Int. J. Product. Res., 44: 4445-4464. DOI: 10.1080/00207540600597138

Ali, M. and X. Zeng, 2010. A novel technique for extraction foetal electrocardiogram using adaptive filtering and simple genetic algorithm. Am. J. Bio., 2: 75-81. DOI: 10.3844/amjbsp.2010.75.81

Al-Taharwa, I., A. Sheta and M. Al-Weshah, 2008. A mobile robot path planning using genetic algorithm in static environment. J. Comput. Sci., 4: 341-344. DOI : 10.3844/jcssp.2008.341.344

Aven, T. and J. Vinnem, 2007. Risk Management: With Applications from the Offshore Petroleum Industry. 1st Edn., Springer Science and Business Media, Berlin, ISBN-10: 1846286522, pp: 200.

Cai, J., X. Liu, Z. Xiao and J. Liu, 2009. Improving supply chain performance management: A systematic approach to analyzing iterative KPI accomplishment. Decis. Supp. Syst., 46: 512-521. DOI: 10.1016/j.dss.2008.09.004

El Dabee, F. and R. Hokoma, 2012. Just-in-time for reducing inventory costs throughout a supply chain: A case study. World Acad. Sci. Eng. Technol., 6: 547-550.

El Dabee, F., R. Marian and Y. Amer, 2013a. A novel optimisation model for simultaneous cost-risk reduction in multi-suppliers just-in-time systems. J. Comput. Sci., 9: 1778-1792. DOI: 10.3844/jcssp.2013.1778.1792

El Dabee, F., R. Marian and Y. Amer, 2013b. Development of a model for simultaneous cost-risk reduction in JIT systems using multi-external and local backup suppliers. Automat. Control Intell. Syst., 1: 42-52. DOI: 10.11648/j.acis.20130103.12
Paksoy, T. and C. Chang, 2010. Revised multi-choice goal programming for multi-period, multi-stage inventory controlled supply chain model with popup stores in guerrilla marketing. Applied Math. Modell., 34: 3586-3598. DOI: 10.1016/j.apm.2010.03.008

Park, Y.J., S. Chun and B. Kim, 2011. Cost-sensitive case-based reasoning using a genetic algorithm: Application to medical diagnosis. Artificial Intell. Medic., 51: 133-145. DOI: 10.1016/j.artmed.2010.12.001

Radhakrishnan, P., V. Prasad and M. Gopalan, 2009. Optimizing inventory using genetic algorithm for efficient supply chain management, J. Comput. Sci., 5: 233-241. DOI: 10.3844/jcssp.2009.233.241

Rawal, R. and K. Inamdar, 2014. Review on various optimisation techniques used for process parameters of resistance spot welding. Int. J. Current Eng. Technol., 3: 160-164.

Scavino, E., D. Wahab, H. Basri, M. Mustafa and A. Hussain, 2009. A genetic algorithm for the segmentation of known touching objects. J. Comput. Sci., 5: 711-716. DOI: 10.3844/jcssp.2009.711.716

Sivanandam, S. and S. Deepa, 2008. Introduction to Genetic Algorithms. Springer Berlin Heidelberg, New York, ISBN-10: 978-3-540-73189-4, pp: 165-209.

Tasan, S. and S. Tunali, 2008. A review of the current applications of genetic algorithms in assembly line balancing. J. Intell. Manufact., 19: 49-69. DOI: 10.1007/s10845-007-0045-5

Torabi, S., S.F. Ghomi and B. Karimi, 2006. A hybrid genetic algorithm for the finite horizon economic lot and delivery scheduling in supply chains. Eur. J. Operat. Res., 173: 173-189. DOI: 10.1016/j.ejor.2004.11.012

Wang, K. and Y. Wang, 2008. Applying genetic algorithms to optimize the cost of multiple sourcing supply chain systems-an industry case study. Springer Berlin Heidelberg, 355-372. DOI: 10.1007/978-3-540-76286-7_16

Wong, W., C. Kwong, P. Mok and W. Ip, 2006. Genetic optimization of JIT operation schedules for fabric-cutting process in apparel manufacture. J. Intell. Manufact., 17: 341-354. DOI: 10.1007/s10845-005-0007-8

Yussof, S., R. Razali and O. See, 2011. An investigation of using parallel genetic algorithm for solving the shortest path routing problem. J. Comput. Sci., 7: 206-215. DOI: 10.3844/jcssp.2011.206.215

Zhu, G. and S. Kwong, 2010. Gbest-guided artificial bee colony algorithm for numerical function optimization. Applied Math. Comput., 217: 3166-3173. DOI: 10.1016/j.amc.2010.08.049