The importance of supply chain and logistics management has been widely recognised. Effective management of the supply chain can reduce costs and lead times and improve responsiveness to changing customer demands. This paper proposes a multi-matrix real-coded Generic Algorithm (MRGA) based optimisation tool that minimises total costs associated within supply chain logistics. According to finite capacity constraints of all parties within the chain, Genetic Algorithm (GA) often produces infeasible chromosomes during initialisation and evolution processes. In the proposed algorithm, chromosome initialisation procedure, crossover and mutation operations that always guarantee feasible solutions were embedded. The proposed algorithm was tested using three sizes of benchmarking dataset of logistic chain network, which are typical of those faced by most global manufacturing companies. A half fractional factorial design was carried out to investigate the influence of alternative crossover and mutation operators by varying GA parameters. The analysis of experimental results suggested that the quality of solutions obtained is sensitive to the ways in which the genetic parameters and operators are set.

**Keywords**—Genetic Algorithm, Logistics, Optimisation, Supply Chain.

**I. INTRODUCTION**

CUSTOMER satisfaction is one of the major issues for any downstream business that requires effective logistics. Effective logistics may be defined as the art of bringing the right amount of the right product to the right place at the right time with minimising costs related within and between all parties and usually refers to supply chain problems. A typical logistics chain commonly (see Fig. 1) involves a network of tiered suppliers producing raw materials, parts, components, subassemblies, assemblies and final products together with distribution centres/warehouses, wholesalers and retailers/customers.

In order to meet customers’ demand, costs related to raw materials, production, holding, multi-stage transportation and fixed operation costs are arising in the logistics chain network.

The design task is to minimise these costs, which can be mathematically formulated and solved using enumerative methods such as Integer Linear Programming [1] or stochastic search techniques (sometimes called meta-heuristics) such as Genetic Algorithm [2], [3]. However, GA proposed by [2] has not been based on matrix and unfortunately produced infeasible offspring, which is omitted from the iterative evolution process within the GA. The work in [3] ignored raw material, manufacturing and holding cost from the objective function and similarly proposed a repair process for rectifying infeasible chromosome.

Genetic Algorithm (GA) has several advantages; multiple directional searches, problem coding instead of decision variables and using stochastic transition rules [4], [5]. It has therefore been widely used to solve production and operation management (POM) problems such as supply chain and logistics [1], production scheduling [6], facility layout [7] and university course timetabling [8]. However, the GA applications on some POM problem areas such as transportation within logistics chain network [9], quality planning, short/long term forecasting and short-term capacity planning have rarely been found [10].

The flexibility of chromosome representation is one of the major advantage strategies within the genetic algorithm (GA). For example, a single row chromosome representation is normally used for solving sequencing or scheduling problems whilst a single matrix-based chromosome representation is required for a candidate solution of a single stage transportation problem. Sun et al. [11] have applied a single matrix-based GA for solving the unit commitment problem, which plays an important role in the economic operation of
power system. In their work, the repair mechanism is additionally embedded in the GA for dealing with the infeasible solutions generated. However, a multi-matrix based chromosome is required to represent a multiple stage transportation problem, which is usually occurred in a logistics chain network.

Genetic operations including crossover and mutation are the main stochastic search process within the GA. Crossover operation helps search strategy to explore the solution space whilst exploitation is conducted by the mutation mechanism. Fifteen crossover operations and eleven mutation techniques have been reviewed and investigated in literature [12]. However, the majority of those operations are suitable for single row chromosome and often produce infeasible offspring. In the present work, alternative crossover and mutation operators were proposed and their performance was investigated.

The objectives of this paper were to i) present the mathematical model for minimising total costs raised from raw material, manufacturing, holding inventory, transporting between parties and fixed operation costs; ii) describe a multi-matrix real-coded Genetic Algorithm (MRGA) that always guarantee feasible solutions obtained from both initialisation and evolution processes and iii) present the computational experiments for investigating genetic parameters and operators using three sizes of dataset.

The remaining of the paper is organised as follows. Section 2 describes the logistics chain network and its related costs including mathematical model. Matrix based Genetic Algorithm that was developed for minimising total costs arising in logistics networks are presented in section 3. Section 4 briefly describes the case study and the data used. Section 5 presents the experimental design and analysis followed by conclusions in section 6.

II. LOGISTICS CHAIN NETWORK (LCN) PROBLEM

Typical problem arising in a logistic chain network is involved in determining the choice of available facilities (such as plants and warehouses) to be opened and in designing the transportation routing between parties (from suppliers via plants and warehouses/distribution centres to customers) in order to meet customers’ demand with minimum cost [13], [14]. The problem is usually constrained by the finite capacity limitation on each suppliers, plants and warehouses. Due to the transportation network design, the LCN problem can therefore be referred to a multiple stages capacitated transportation/ allocation problem known to be NP-hard, which can alternatively be solved using GA [5].

Considering a general steady stage supply chain network, there are a number of suppliers, plants, distribution centres/warehouses and its capacity limitation aiming to satisfy customer demand given by the contract. For example in capital goods companies, most of main products are high value and are demanded in low volume [15]. The desired amount of goods are manufactured and delivered based on JIT philosophy. The objective function (1) reflects a total costs to be minimised. The equation composed of five parts. The first three parts considers the raw materials, manufacturing, holding costs and its transportation between parties, respectively. The rest two parts take into account the fixed operating cost for plants and warehouses.

\[
\text{Minimise } \sum_{i=1}^{L} \sum_{j=1}^{J} (a_{ij} + r_{i})x_{i,j} + \sum_{j=1}^{J} \sum_{k=1}^{K} (b_{jk} + m_{j})y_{j,k} + \sum_{k=1}^{K} \sum_{l=1}^{L} (c_{kl} + h_{k})z_{k,l} + \sum_{j=1}^{J} \sum_{f=1}^{F} f_{j}t_{j} + \sum_{k=1}^{K} \sum_{l=1}^{L} f_{k}t_{k} \tag{1}
\]

\[
\sum_{j=1}^{J} x_{i,j} \leq S_{i} \quad \forall i,j \tag{2}
\]

\[
\sum_{j=1}^{J} y_{j,k} \leq P_{j} \quad \forall j,k \tag{3}
\]

\[
\sum_{j=1}^{J} z_{i,j} \leq W_{k} \quad \forall i,k \tag{4}
\]

\[
\sum_{l=1}^{L} z_{i,j} = C_{l} \quad \forall k,l \tag{5}
\]

\[
\sum_{j=1}^{J} x_{i,j} \leq P_{j}t_{j} \quad \forall j \tag{6}
\]

\[
\sum_{j=1}^{J} y_{j,k} \leq W_{k}t_{k} \quad \forall j,k \tag{7}
\]

\[
\sum_{j=1}^{J} x_{i,j} = \sum_{j=1}^{J} y_{j,k} = \sum_{k=1}^{K} z_{i,j} = \sum_{l=1}^{L} C_{l} \quad \forall i,j,k \tag{8}
\]

\[
x_{i,j}, y_{j,k} , z_{i,j} \geq 0 \quad \forall i,j,k \tag{9}
\]

\[
t_{j}, t_{k} = \{0, 1\} \quad \forall j,k \tag{10}
\]
Constraint (2) (3) (4) and (5) represent the capacity limitations of the suppliers, plants, warehouse/distribution centres and customers, respectively. Constraint (6) ensures that raw materials are delivered to only operating plants; likewise, constraint (7) for only operating warehouses. Constraint (8) ensures that the same amount of items is transported in each stage and also meets customers’ demand. In the case of unbalanced supply and demand, a dummy supplier or customer may be introduced. Constraint (9) ensures that transportation variables are greater or equal to zero whilst binary decision variables are specified in the last constraint.

III. MULTI-METRIX REAL-CODED GENETIC ALGORITHM (MRGA)

Genetic Algorithm (GA) is stochastic search technique based upon the mechanics of natural selection [4], [5]. The basic idea came from an analogy with biological evolution, in which the fitness of individual determines its ability to survive and reproduce. In this work, multi-matrix real-coded Genetic Algorithm (MRGA) was developed for minimising total costs arising in the logistics chain network. The initialisation process and genetic operations that always produce feasible chromosome were proposed. The general process of the MRGA that mainly included chromosome representation and initialisation, genetic operations and chromosome evaluation and selection are described in followings sub-sections.

A. Chromosome Representation and Initialisation

Multiple matrix based chromosome representation is used to represent the multi-stage transportation network between parties in the logistics chain network (LCN). For example, three stages LCN problem (shown in Figure 1) consists of a set of suppliers (I), plants (J), warehouses/distribution centres (K) and customers (L). This gives raise three-stage transportation matrices (M) of suppliers to plants (M_{i,j}), plants to warehouses (M_{j,k}) and warehouses to customers (M_{k,l}). Each matrix is called sub-chromosome. Fig. 2 shows multiple matrix based chromosome representation.

\[
\begin{bmatrix}
  x_{11} x_{12} \ldots x_{1J} \\
  x_{11} x_{12} \ldots x_{1K} \\
  \vdots \\
  x_{11} x_{12} \ldots x_{1L}
\end{bmatrix}
\begin{bmatrix}
  x_{21} x_{22} \ldots x_{2J} \\
  x_{21} x_{22} \ldots x_{2K} \\
  \vdots \\
  x_{21} x_{22} \ldots x_{2L}
\end{bmatrix}
\begin{bmatrix}
  \vdots \\
  \vdots \\
  \vdots \\
  \vdots
\end{bmatrix}
\begin{bmatrix}
  x_{11} x_{12} \ldots x_{1J} \\
  x_{11} x_{12} \ldots x_{1K} \\
  \vdots \\
  x_{11} x_{12} \ldots x_{1L}
\end{bmatrix}
\]

Fig. 2 Multiple matrix real-coded based chromosome representation

In the present work, the chromosome initialisation process, in which all chromosomes generated were ensured to be feasible solutions, was proposed. The process sequentially considered matrix by matrix based on the finite capacity limitations arising from all parties. The process of sub-chromosome initialisation was divided into two parts; creating sequence number (s/n) and assigning the values for each element (x_{ij}) in the matrix size of IxJ for instance.

**Part I: Creating s/n for all elements (x_{ij}) in the matrix (M_{i,j}).**

Step 1 Generate random value (v_{ij}) between 0 to 1 for all elements (x_{ij}) in M_{i,j}.

Step 2 Find an ascending sequence starting from 1 to IxJ for all x_{ij} by considering the value of v_{ij}. The x_{ij} with smallest value of v_{ij} will be assigned a sequence number (s/n) = 1 whilst s/n of IxJ will be assigned to the x_{ij} which has the largest value of v_{ij}.

**Part II: Assigning the values of x_{ij} in the matrix (M_{j,k}).**

Step 1 Set the values of all x_{ij} initially equal to zero.

Step 2 Start from the element x_{ij} with s/n = 1, then repeat the following steps until s/n = IxJ.

Step 3 Compare the capacity constraints of row i^{th} (r_{i}) and column j^{th} (c_{j}). If r_{i} ≤ c_{j}, then x_{ij} = x_{ij} + r_{i}; c_{j} = c_{j} - r_{i}; and set r_{i} = 0. Otherwise, x_{ij} = x_{ij} + c_{j}; r_{i} = r_{i} - c_{j}; and set c_{j} = 0.

Step 4 Then increasing the sequence number; s/n = s/n + 1.

B. Genetic Operations: Crossover and Mutation

Several crossover and mutation operations have been intensively reviewed and statistically investigated in literature [12]. Unfortunately, most of them are not matrix based operations and do not guarantee feasible offspring regarding to the constraints considered. There are three ways to deal with infeasible chromosomes: i) discarding them; ii) applying a high penalty in the fitness function so that they are unlikely to survive; or iii) repairing them [16]. Avoiding infeasible solutions may be benefit on decreasing the iterative computational effort. In this present work, two matrix based crossover and mutation operations that always guarantee feasible offspring were developed and described as follows:

Crossover type I was based on the concept of one point crossover by performing between matrices. For example, two chromosomes were randomly selected as parents (see Fig. 3), each of which consisting of three sub-chromosomes (matrices); M_{i,j}, M_{j,k} and M_{k,l}. A cutting point was randomly generated between matrices and then performing a swap.

\[
\begin{bmatrix}
  x_{11} x_{12} \ldots x_{1J} \\
  x_{11} x_{12} \ldots x_{1K} \\
  \vdots \\
  x_{11} x_{12} \ldots x_{1L}
\end{bmatrix}
\begin{bmatrix}
  x_{21} x_{22} \ldots x_{2J} \\
  x_{21} x_{22} \ldots x_{2K} \\
  \vdots \\
  x_{21} x_{22} \ldots x_{2L}
\end{bmatrix}
\begin{bmatrix}
  \vdots \\
  \vdots \\
  \vdots \\
  \vdots
\end{bmatrix}
\begin{bmatrix}
  x_{11} x_{12} \ldots x_{1J} \\
  x_{11} x_{12} \ldots x_{1K} \\
  \vdots \\
  x_{11} x_{12} \ldots x_{1L}
\end{bmatrix}
\]

Fig. 3 Type I and II of crossover operations

Crossover type II was aimed to perform crossover operation on a randomly selected matrix (see Fig. 3). The similar concept described in the chromosome initialisation process was adopted as follows:

Step 1 Randomly select a point regarding to the length of the sub-chromosome.

Step 2 Perform one point crossover operation on the
sequence number \((s/n)\) created during chromosome initialisation for each element \((x_i)\) in the corresponding matrix \((M_{ij})\). This step is therefore reproducing two offspring that have new sequence number for each element \((x_i)\) in the matrix.

**Step 3** Follow the process of assigning the values of \(x_i\) as described in part 2 of the chromosome initialisation procedure for all offspring obtained from step 2.

**Procedure: Mutation type I.**

**Step 1** Randomly select a sub-chromosomes in a parent.

**Step 2** Randomly choose a gene within the selected sub-chromosome and then perform a swap of sequence number \((s/n)\) between the chosen gene with the successive gene. This step is therefore reproducing an offspring that have new sequence number for each element \((x_i)\) in the matrix.

**Step 3** Perform the process of assigning the values of \(x_i\) in the matrix described in part 2 of chromosome initialisation procedure for the offspring obtained from step 2.

**Procedure: Mutation type II.**

**Step 1** Randomly select a sub-chromosomes in a parent.

**Step 2** Perform part 1 and 2 described in the chromosome initialisation procedure for the offspring obtained from step 2. This means that a brand new matrix replaces the chosen one.

**C. Chromosome Evaluation and Selection**

Chromosome evaluation is usually applied to measure the performance (fitness value) of a candidate solution (individual) by determining an objective (fitness) function. The higher fitness value of individual is, the higher its chances to be selected onto the next generation. In this present work, the total costs arising within and between parties in the logistics chain network described in section 2 was used as fitness function. The famous chromosome selection called roulette wheel was then applied for randomly choosing the same amount of individual onto the next generation. The MRGA process was repeated until the termination criteria were satisfied.

**IV. CASE STUDY**

The logistic chain network (LCN) problem is to determine the choice of available facilities (plants and warehouses) to be opened and to design the transportation routing between parties, each of which has finite resource capacity, in order to meet customers’ demand with minimum cost. In this present work, three problem sizes (see Table I) for designing the LCN were proposed in order to test the model developed. For example, the large problem involved eight suppliers, sixteen plants, sixteen warehouses and eight retailers. To solve the problem using linear programming (LP), it required 512 integer and 32 binary variables based on 625 constraints. The best solutions for these benchmarking problems were initially identified by LP method using a software package. Due to the limited (student) version of the software package used, only total costs for small and medium size problems were found at 87,500 and 187,800 baht, respectively. These results were then used for benchmarking the performance of the model developed in the next section.

**TABLE I**

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Small</th>
<th>Medium</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of suppliers</td>
<td>4</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Number of plants</td>
<td>6</td>
<td>10</td>
<td>16</td>
</tr>
<tr>
<td>Number of warehouses</td>
<td>6</td>
<td>10</td>
<td>16</td>
</tr>
<tr>
<td>Number of retailers</td>
<td>4</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Total number of integer variables</td>
<td>84</td>
<td>260</td>
<td>512</td>
</tr>
<tr>
<td>Total number of binary variables</td>
<td>12</td>
<td>20</td>
<td>32</td>
</tr>
<tr>
<td>Number of constraints</td>
<td>129</td>
<td>337</td>
<td>625</td>
</tr>
</tbody>
</table>

**V. EXPERIMENTAL DESIGN AND ANALYSIS**

A two-step sequential experiment was adopted in this present work. The first experiment (Experiment A) was aimed to investigate the influence of alternative crossover and mutation operators by varying GA parameters. Another experiment (Experiment B) was intended to compare the performance of the developed model with the benchmarking results obtained from enumerative method called linear programming.

**A. Experiment A**

Half fractional factorial experimental design [17] with ten replications was carried out to solve each problem size. The experimental factors and its values considered are shown in Table II. The first factor was the combination of population size and number of generations (P/G), which determines the total chromosomes to be investigated. This factor had an influence on the exploration process of seeking (generated) results in the solution space and also delaying the execution time of the computational run. In this present work, total amount of chromosomes generated was fixed at 1,000 according to preliminary test runs. The values setting of the probabilities of crossover (Pc) and mutation (Pm) was based on the suggestions in previous research [18], [19]. The remaining two factors were the crossover and mutation operators (COP and MOP) described in the aforementioned section. These alternative operators proposed in this work were matrix based operations and always guarantee to produce feasible offspring.

**TABLE II**

<table>
<thead>
<tr>
<th>Factors</th>
<th>Levels (coded)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population/Generation (P/G)</td>
<td>Low (-)</td>
</tr>
<tr>
<td>Probability of crossover (Pc)</td>
<td>0.6</td>
</tr>
<tr>
<td>Probability of mutation (Pm)</td>
<td>0.1</td>
</tr>
<tr>
<td>Crossover operation (COP)</td>
<td>Type I</td>
</tr>
<tr>
<td>Mutation operation (MOP)</td>
<td>Type I</td>
</tr>
<tr>
<td>Type II</td>
<td></td>
</tr>
</tbody>
</table>

ISNI:0000000091950263
The experiments for each problem size were carried out on a notebook computer with AMD Athlon 1400+ and 128 MB SDRAM. The experimental results obtained from 160 (24x10) runs were analysed using a general linear form of analysis of variance and main effect plots. The significant factors indicated using underline and its appropriate setting based on each problem size is summarised in Table III. It can be seen that all factors with an exception of P can be statistically significant with 95% confidence interval. The best settings of some factors (P/G, Pm and COP) were in agreement for all problem size but the remaining factors were not. These findings of parameters’ setting were used in the sequential experiment presented in the next section.

### TABLE III

<table>
<thead>
<tr>
<th>Problem sizes</th>
<th>GA parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P/G</td>
</tr>
<tr>
<td>Small</td>
<td>Low</td>
</tr>
<tr>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Large</td>
<td>Low</td>
</tr>
</tbody>
</table>

B. Experiment B

This experiment was aimed to compare the results obtained from the Multi-matrix Real-coded Genetic Algorithm (MRGA) with the benchmarking results obtained from linear programming (see Table IV). It was found that the total costs obtained from the MRGA were very close to the optimum solutions identified by linear programming using a software package. Due to the limited (student) version of the software used, only optimum solutions of small and medium problem were provided. The large problem required 544 variables and 625 constraints, which exceed the limitation of the version.

### TABLE IV

<table>
<thead>
<tr>
<th>Problem sizes</th>
<th>Total cost (Baht)</th>
<th>% near optimum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MRGA</td>
<td>Optimum</td>
</tr>
<tr>
<td>Small</td>
<td>88,150</td>
<td>87,500</td>
</tr>
<tr>
<td>Medium</td>
<td>199,000</td>
<td>187,800</td>
</tr>
<tr>
<td>Large</td>
<td>674,300</td>
<td>-</td>
</tr>
</tbody>
</table>

VI. CONCLUSION

This paper proposed the Multi-matrix Real-coded Genetic Algorithm (MRGA) based optimisation tool that minimises total costs associated within supply chain logistics. Since the simple GA often produces infeasible chromosomes during initialisation and evolution processes due to finite capacity constraints of all parties within the chain, an alternative matrix based chromosome initialisation procedure, crossover and mutation operations that always guarantee feasible solutions were therefore proposed in this paper. The proposed algorithm was tested using three sizes of benchmarking dataset of logistic chain network, which are typical of those faced by most global manufacturing companies. A sequential experiment was systematically carried out to investigate the influence of alternative crossover and mutation operators by varying GA parameters and to identify how close the MRGA can find the near optimum solutions. The analysis of experimental results suggested that the quality of solutions obtained is sensitive to the ways in which the genetic parameters and operators are set. It was also found that the best results obtained from the MRGA were less than 6 percent deviated from the optimum solutions.

REFERENCES

Pupong Pongcharoen is an Assistant Professor in Industrial Engineering, Faculty of Engineering, Naresuan University, Phitsanulok, Thailand. He graduated a bachelor degree in industrial engineering from Chiang Mai University, Chiang Mai, Thailand. He graduated a master degree from the Department of Industrial System Engineering and Management, School of Advanced Technology, the Asian Institute of Technology. He received a Ph.D. from the Department of Materials, Mechanical and Manufacturing Engineering, Faculty of Engineering at the University of Newcastle upon Tyne, Newcastle upon Tyne, United Kingdom.

He has been appointed as a lecturer in Industrial Engineering Department, Naresuan University since 1994. His teaching and research interests include manufacturing planning and scheduling especially in engineer/make-to-order strategy, computer simulation, supply chain and logistics management, metaheuristics, applied statistics and operations research on industrial systems. He has had several publications in international journals such as Journal of Applied Statistics, International Journal of Production Economics, International Journal of Technology Management, European Journal of Operations Research and Scienceasia.


Aphirak Khadwilard is a lecturer in the Faculty of Engineering, Rajamangala University of Technology Lanna, Tak, Thailand. He graduated a bachelor degree in mechanical engineering from Rajamangala Institute of Technology, Bangkok, Thailand. He graduated a master degree from the Department of Industrial Engineering, Faculty of Engineering, Naresuan University, Phitsanulok, Thailand.

He has been appointed as a lecturer in the Faculty of Engineering, Rajamangala University of Technology Lanna since 1995. His teaching and research interests include automatic control, mechanical vibration and application of metaheuristics in engineering design.

Anothai Klakankhai is currently a staff in the Faculty of Engineering, Naresuan University, Phitsanulok, Thailand, from which he graduated bachelor and master degrees in industrial engineering.

His research interests include supply chain and logistics management and metaheuristics applied on industrial systems.