An Immune Genetic Algorithm for Inter-Cell Layout Problem in Cellular Manufacturing System

ABSTRACT

The objective function of inter-cell layout problem minimizes the total inter-cellular material handling cost. It is mostly significant with moderate production quantity in Cellular Manufacturing Systems (CMS). This problem is classified as Quadratic Assignment Problem (QAP) which is NP-Hard in nature. Heuristic techniques are extremely effective for such problems. In this paper we have proposed a novel Immune Genetic Algorithm (Immune-GA-RS) to obtain competent inter-cell layout in the vicinity of CMS. It exploits an elitist replacement strategy in order to improve the rate of convergence. The proposed method is successfully tested with 8 datasets which are being widely used for inter-cell layout design problems. Proposed Immune-GA-RS is compared with two variants of the Genetic Algorithms, GA-RS and Alt-GA-RS and 7 other published layout design techniques. Results portray that Immune-GA-RS acquires 11.11% improved solutions with 7.72% reduced CPU time on an average. Further Immune-GA-RS is tested on 36 structured QAP instances available through QAPLIB and shown to outperform other two GA variants while attaining optimal solutions for 33 instances and also shown to outpace five recent QAP based algorithms, SA (S&S-2008), GA (S&S-2010), SC-TS (F&M-2011), IHGA (Mis-2004), GA-(KTF&D-2011) while attaining smaller solution gap for 11 test instances and obtain at least equal or better quality solutions for 24 instances. We conclude our work with a statistical data test to signify the results of Immune-GA-RS.

Keywords: Inter-cell layout; Material handling; Intercell material flow; Immune Genetic algorithm; Quadratic assignment problem.

1. Introduction

Group Technology (GT) is a manufacturing philosophy, has been effectively employed to reduce the output time of production systems, material handling costs, the work-in-progress and stocks of finished goods, etc. GT enhances the competence of the system by improving the forecast precisions in uncertain production scenarios (Selim, Askin, and Vakharia, 1998; Won and Currie, 2007). Cellular manufacturing (CM) is a function of GT and has been evolved as a potential replacement of the traditional manufacturing systems. CM could be illustrated as a hybrid system which exploits the advantages of jobshop (variety) and flowshop (higher rate of production) production strategies. To design an efficient CMS, first we have to decompose the manufacturing system into several cells by recognizing and exploiting the resemblances amongst parts and machines, next we need to design efficient layout to control the material handling costs. The 1st step is identified as the classical cell formation problem (CFP). CFP obtains machine-cells and corresponding part families and minimizes the inter-cell part travels. The cells are not practically independent to each other since some of the operations of a part might be required to be performed beyond its granted cell (Heragu and Kakuturi, 1997). That part is distinguished as an 'exceptional element'. The machines which process that part and do not belong to the dedicated cell of that part are distinguished as the 'bottleneck' machines. These machines are solely responsible for inter-cell material flow. The newly formed cells are believed to be assigned to optimal locations inside the factory area to minimize inter-cell material flow. This assignment problem is classified as cell layout problems (CLP) or inter-cell layout problems (Kulkarni and Shankar, 2007). A competent layout does not only improve the performance of the system but also

minimizes nearly 40-50% of the production costs (Balakrishnan and Cheng, 2009). However layout design in CMS has not acquired much interest of researchers in recent past since most of the relevant works investigate the CFPs (Wang and Sarker, 2002). In the domain of CMS only some articles particularly considered CLPs as independent research problems. Logendran (1991) first proposed a mathematical model that considered the operational sequences to evaluate the inter-cell and intra-cell moves and the impact of the cell layout to illustrate the inter-cell material flow. Alfa, Chen and Heragu (1992) suggested a concurrent sub-optimal solution of the machine grouping and layout problem in CMS using a simulated annealing (SA) approach. Sarker and Yu (1994) reported a twofold procedure for duplicating bottleneck machines in CMS and solved a cell layout problem which minimizes the inter-cell material flow. Tang and Abdel-Malek (1996) stated a flow-network-oriented inter-cell layout model in three key steps: (1) K-Shortest Path method to reduce various flows of a system; (2) a flow pattern which designates the system's aisle structure; (3) cell allocation around the flow pattern and the aisle structure within a limited area floor plan. Lee (1998) adopted the intra-cell and inter-cell layout design problems using a three-phase interactive method following the decomposition strategy to reduce the large problem into smaller sub-problems with minute details. Salum (2000) introduced some similarity measures to construct an intra-cell layout by placing the machines with higher similarity value next to each other to minimize the material handling time of the system. Wang and Sarker (2002) stated a lower bound on the inter-cell layout problem (QAP) and prescribed a 3-pair comparison heuristic and a 'bubble search' technique to minimize the inter-cell material flow incurred due to bottleneck machines. Chan, Chan and Kwong (2004) proposed the MAIN algorithm to solve the intra-cell layout problems in static and dynamic conditions by considering part-handling factor, machine rearrangement cost and machine closeness factor. Solimanpur, Vrat and Shankar (2004) developed ant colony algorithm (ACO) to solve the QAP model of inter-cell layout and compared their results with other techniques such as H63, HC63-66, CRAFT and bubble search successfully. Wu et al. (2006) implemented a new genetic algorithm (GA) to solve the cell design and group layout problems concurrently incorporating some important production factors such as operational sequences, part demand, transfer batch, machine capacities, and layout types. Chan et al. (2006) proposed a GA based algorithm to solve the layout based QAP along with a cell formation model considering the linear layout shape. Kulkarni and Shankar (2007) employed a GA to the inter-cell layout problem and validated the performance of GA with well-known layout design techniques. Tavakkoli-Moghaddam et al. (2007) presented a new mathematical model to solve a layout problem with varying demand to minimize the total layout costs. Mahdavi and Mahadevan (2008) developed an algorithm that concurrently obtains manufacturing cells and the intra-cell layout successfully. Ahi et al. (2009) applied a TOPSIS based initial solution generating method for order preference by similarity to the ideal solution that leads to determination of cell formation, intra-cell and inter-cell layouts and shown further improvements to the proposed method. Ariafar and Ismail (2009) proposed a new QAP model for inter-cell and intra-cell layouts and solved that using an SA algorithm efficiently. Ma and Zhang (2010) demonstrated the dynamic layout framework based on the reconfigurable CMS aiming at the enterprise problems. It solves CFP and CLP jointly using alternative process routes and multiple machine types available for the operations. Jolai, Taghipour and Javadi (2011) employed a binary particle swarm algorithm (PSO) to solve a QAP model for inter-cell and intra-cell layout problems considering uncertain demand of parts and batch sizes using a variable neighborhood search. Leno et al. (2011) discussed the multi-objective CLP with unequal size of cells which minimizes the total material handling cost in the first place and subsequently maximizes the distance-weighted closeness factor of cells and solved it using a GA. Arkat, Farahani and Hosseini et al. (2011) employed two techniques based on GA to solve an

integrated model of cell formation, cell layout and cell scheduling. Similar issues are addressed by Kia et al. (2012) by developing a novel non-linear model and solved that using an SA algorithm and compared successfully with the solutions of Lingo software.

Table 1. Summary of the literature review done on cellular layout problems

References	Solution Methodology Used	Scope of the layout problem considered		Objectives considered (either minimization type (total cost) or maximization type (profit/closeness/similarity/machine utilization))
		Inter-Cell	Intra-Cell	
Logendran (1991)	Heuristic based on operational sequence and configuration	×	×	Total inter-cell and intra-cell moves and utilization of workstations
Alfa, Chen and Heragu (1992)	SA	×	×	Total flow of material (combined form of cell formation and layout design)
Sarker and Yu (1994)	Two phase heuristic method	×		Inter-cell material flow and bottleneck machine that need to be duplicated
Tang and Abdel-Malek (1996)	A flow network oriented heuristic	×		Cell to cell material flow considering a flow pattern within strict floor plan
Lee (1998)	Three phase heuristic technique	×	×	Total inter-cell and intra-cell material handling cost
Salum (2000)	two-phase method based on total manufacturing lead time reduction		×	total manufacturing lead time (MLT) reduction and intra-cell flow
Wang and Sarker (2002)	3-pair comparison and bubble search method	×		Total inter-cell material flow
Chan, Chan and Kwong (2004)	MAIN (Machines Allocation Inter-relationship) algorithm		×	Material handling cost and rearrangement cost
Solimanpur, Vrat and Shankar (2004)	ACO	×		Total inter-cell material flow
Wu et al. (2006)	Hierarchical GA	×	×	Total flow of material (combined form of cell formation and layout design)
Chan et al. (2006)	Macro approach based 2-stage heuristic technique	×		Total inter-cell material flow (combined form of cell formation and layout design)
Kulkarni and Shankar (2007)	GA	×		Total inter-cell material flow
Tavakkoli-Moghaddam et al. (2007)	Branch and Bound method	×	×	Total flow of material
Mahdavi and Mahadevan (2008)	CLASS (Cell and Layout Solution using Sequence data) algorithm		×	Total intra-cell material flow (combined form of cell formation and layout design)
Ahi et al. (2009)	TOPSIS based 2-stage Heuristic method	×	×	Total flow of material (combined form of cell formation and layout design)
Ariafar and Ismail (2009)	SA	×	×	Total inter-cell and intra-cell material handling cost
Ma and Zhang (2010)	Dynamic Programming	×	×	Total flow of material (combined form of cell formation and layout design)
Jolai, Taghipour and Javadi (2011)	variable neighborhood binary PSO	×	×	Total inter-cell and intra-cell material flow
Arkat, Farahani and Hosseini (2011)	GA	×	×	Total flow of material (combined form of cell formation and layout design considering cell scheduling decisions)
Leno et al. (2011)	GA	×		Total flow of material, total distance-weighted closeness rating and penalty to force the solutions to satisfy floor boundary condition
Kia et al. (2012)	SA		×	Total intra-cell material flow (combined form of cell formation and layout design considering cell scheduling decisions)

The layout problem in CMS is classified as the QAP which is believed to be NP-hard in nature (Kaufman and Broeckx, 1978). Heuristic or meta-heuristic approaches are extremely effective for such problems and accomplish near optimal solutions (Solimanpur, Vrat, and Shanker, 2004). Genetic Algorithms are mostly practised among all the meta-heuristic approaches for the solution of combinatorial optimization problems (Anderson and Ferris, 1994). GA based methods are heavily employed in the domain of CMS to design efficient cells. A comprehensive review of GAs in CMS can be found in Ghosh et al. (2010). Ahuja, Orlin and Tiwari (2000), Fleurent and Ferland (1994), and Tate and Smith (1995) used GAs as the solution approaches to QAP. Drezner (2003) proposed a new GA based QAP solving technique with problem specific improved crossover method and a Tabu Search approach. Ji, Wu and Liu (2006) reported a hybrid GA based technique to examine the solvability of QAP instances. Wu and Ji (2007) proposed a GA based on a new replacement strategy which aims to improve the performance of the method. Kratica et al. (2011) proposed a novel genetic encoding scheme for QAP which is implemented with objective function and modified genetic operators. We have summarized the outcomes of the literature review in Table 1. Following facts can be stated henceforth,

a) Most of the articles have considered a combined form of the layout and CFPs which may perhaps complicate the model unnecessarily while the CLPs could be addressed independently.

b) Few researchers have proposed the QAP model as a combined form of inter-cell and intra-cell layout problems. However the inter-cell and intra-cell layout subject matters might be discussed individually due to the consequence of critical and intrinsic factors related to CMS.

c) Single and multi-period layout problems are not distinctly and prominently discussed in the CMS literature. Both of these two problems have enormous scopes to be explored.

d) Most of the researchers considered problem specific heuristic approaches due to the combinatorial nature of the layout problems. Few of the articles proposed metaheuristic techniques such as SAs, ACOs and GAs for the CLPs with prominent solutions. However there could be many other efficient methodologies which can solve CLP successfully.

In this paper we have developed a novel immunity based genetic algorithm (Immune-GA-RS) for the CLP. The technique exploits elitist replacement strategy which improves the convergence rate drastically. Further we have developed two different variants of GA and compared the results obtained by these three techniques. The rest of the article is structured in the following manner, in §2 we have introduced the QAP formulation of the problem, in §3 we have illustrated the proposed Immune-GA-RS. The simulation study and results are conferred in §4. Lastly, in §5 we have concluded our research study.

2. Problem Definition

Notations:

j, l indices of locations ($j, l = 1, 2, 3\ldots, N$)
i, k indices of cells ($i, k = 1, 2, 3, \ldots, N; f_{ik} = f_{ki}, i \neq k$)
d_{jl} distance between two locations j and l
f_{ik} the amount of material flow from cell i to k

fr_{ik} the inter-cell trips between cell i and k, ($fr_{ik} = f_{ik} + f_{ki}$)

x_{ij} the decision variable ($x_{ij} \in \{0, 1\}$).

We have considered a QAP based inter-cell layout problem in this paper as suggested by Wang and Sarker (2002).
The suppositions of the proposed model are,

- Manufacturing cells are already developed and cell formation solutions are available.
- Size of each cell is equal and the shape and space of the floor area is not limited.
- Distance between two locations is measured from one centre of location to another.
- Material flow from one cell to another is measured beforehand.

The QAP formulation of the proposed MPCLP model is depicted as follows,

$$Minimize\ f(x) = \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{l=1}^{N} fr_{ik} d_{jl} x_{ij} x_{kl} \quad (1)$$

Subject to,

$$\sum_{i=1}^{N} x_{ij} = 1 \qquad j \in N \quad (2)$$

$$\sum_{j=1}^{N} x_{ij} = 1 \qquad i \in N \quad (3)$$

$$x_{ij} = \begin{cases} 1, & if\ cell\ i\ is\ assigned\ to\ location\ l \\ 0, & Otherwise \end{cases} \quad (4)$$

Equation (1) minimizes the total inter-cell material handling costs. Equation (2) and (3) are the assignment constraints. These ensure that each cell is assigned to only one location and each location contains only one cell respectively. Remaining constraint (4) is the 0-1 decision variable x_{ij}.

3. Research Methodology

Immune Genetic algorithm is a stochastic search and optimization technique based on GA and initiated on the basic insights of natural selection (Darwin, 1929), population genetics (Fisher, 1930) and theory of immunity (Chen, 1980). Jiao and Wang (2000) stated an Immune-GA and made this algorithm accustomed among researchers. It is a modified form of standard GA. Standard GA executes iteratively on a set of encoded chromosomes called a population using three basic genetic operators: selection, crossover and mutation (Holland, 1975; Goldberg, 1989). A comprehensive theory of GA can be obtained in the book written by Gen and Cheng (2000). The biological immune system is moderately complicated in nature. It facilitates in shielding against pathogenic organisms. The function of an immune system is to protect our physique from antigens. The idea of immunity can be incorporated in GA by choosing proper 'vaccination' rule and immune selection strategy. The 'vaccination' modifies genes of the current chromosome by altering few bits depending upon the problem so as to conceivably move to the more prominent area of the state space. The immune selection strategy picks the better chromosomes for next stage by applying annealing

schedule. We used the 'vaccination' to retain good genes and adjust other genes. In vaccination, a random number *r* is used to decide whether to pick forward or backward operator. In the next subsections we discuss the Immunity based GA with elitist replacement strategy (Immune-GA-RS) in details.

3.1. Immune-GA-RS

The proposed technique starts with an initial population of randomly selected feasible solutions which are encoded on a string of finite length depending upon the number of cells in the layout. Immune-GA-RS is real coded and depends on the nature of the layout solution.

3.1.1. Initial Population and Encoding Scheme

Initial population is generated randomly following the rule,

a) Every cell is placed in one location, thus there would not be a repeat of string element (gene). Encoding of a layout for a period can be presented as,

5	4	1	3	6	2

This chromosome is based on a 6 cells, 6 locations problem. It implies cell 5, 4, 1, 3, 6, 2 are assigned to location 1 2, 3, 4, 5, and 6 respectively. It is known as a chromosome of Immune-GA-RS. This arrangement is considered as a solution for a (6×6) problem.

3.1.2. Fitness Function

The fitness function essentially evaluates a solution string by computing a numerical score. The CLP objective function (eq. (1)) is used for this purpose. Since the CLP is a minimization type function, therefore the low score of a solution string is desirable in this context.

3.1.3. Selection method

Fitness-proportionate selection scheme frequently forces the algorithm to "exploit" the good areas at the cost of investigating the other parts of the solution space. At the later stage, when all the solution strings of the population are relatively similar (trivial variation in fitness), selection becomes insignificant and the evolution stops with premature convergence (Mitchell, 1999). To avoid such occurrence, the 'sigma truncation' method is adopted as a selection method (Goldberg, 1989). It eventually maintains a robust selection pressure throughout the direction of the execution. Unprocessed fitness values are first transformed into the expected values. The formula used in this work is,

$$E(i,t) = \begin{cases} 1 + \dfrac{f(i) - f_{\mu}(t)}{C_{\sigma} \times f_{\sigma}(t)} & if\ f_{\sigma}(t) \neq 0 \\ 1 & if\ f_{\sigma}(t) \neq 0 \end{cases} \quad (5)$$

E(i, t) is the expected value of individual *i* at time *t*, *f(i)* is the fitness of *i*, $f_{\mu}(t)$ is the mean fitness of the population at time *t*, and $f_{\sigma}(t)$ is the standard deviation of the population fitnesses at time *t*, C_{σ} is the sigma truncating coefficient. A higher value of C_{σ} (=2) can reduce the fitness pressure of the population. This phenomenon would help the better solution to stand out more in the later stage of execution when the population is likely to converge and standard deviation is comparatively low in order to continue evolution.

3.1.4. Implementation of Reproduction Operator

Four genetic operations are employed in Immune-GA-RS implementation. These are reproduction, crossover, mutation and immune operators (vaccination). The reproduction operation triggers best fit chromosomes from the current population and put them into a list called 'elite list' or 'mating pool' to be used for next operations in the evolution. If the current population contains *n* chromosomes and the reproduction rate is defined by *Pr*, then $n \times Pr$ best chromosomes are required to be reproduced and to be put in 'elite list'. Thereafter $(n - n \times Pr)$ chromosomes are replicated from the list of 'elite' chromosomes and added to the 'mating pool'.

3.1.5. Implementation of Crossover Operator

The crossover operator exchanges genetic features between two parent chromosomes selected randomly from 'mating pool' and then produces offsprings or child chromosomes. If the 'mating pool' contains *n* chromosomes and the crossover rate is *Pc*, then $n \times Pc$ chromosomes randomly chosen for crossover. The crossover method applied in this Immune-GA-RS is based on *two-point* operation. An example of the crossover operation is depicted in Figure 1. The crossover points are chosen randomly by generating two random integers *r1* and *r2* between 0 and *m* (*m* is the number of cells in the layout) and then exchanging the $r1^{th}$ to $r2^{th}$ substring of each of the parents. As shown in Figure 1, say $r1 = 3$ and $r2 = 5$ thus 3^{rd} to 5^{th} genes are transferred between parents and two offsprings are generated.

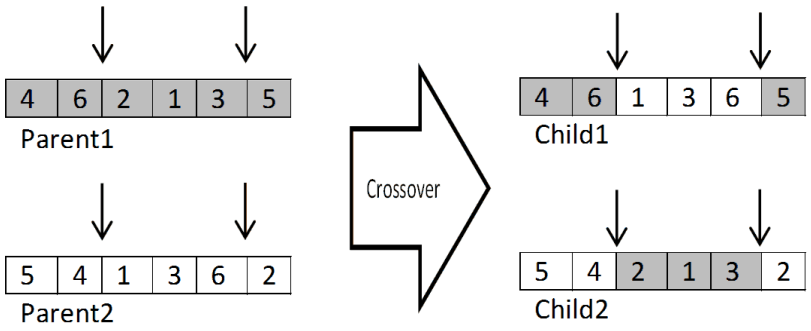

Figure 1. Crossover operation for (6×6) test problem

At this moment in each of the offsprings cell numbers are duplicated which is not desirable. In order to avoid such situation a small heuristic procedure is introduced to restructure the offsprings. The procedure is furnished as,

```
for j=1:r1-1
   for k=r1:r2-1
      if child1(j)==child1(k)
        child1(j)=parent1(k);
      end
        if child2(j)==child2(k)
          child2(j)=parent2(k);
        end
   end
end
for j=r2:m
   for k=r1:r2-1
      if child1(j)==child1(k)
        child1(j)=parent1(k);
```

```
    end
    if child2(j)==child2(k)
      child2(j)=parent2(k);
    end
  end
end
```

This procedure would eventually retain the exchanged genetic structure of the offsprings while avoiding the gene recurrence. The resulting offsprings are portrayed in Figure 2.

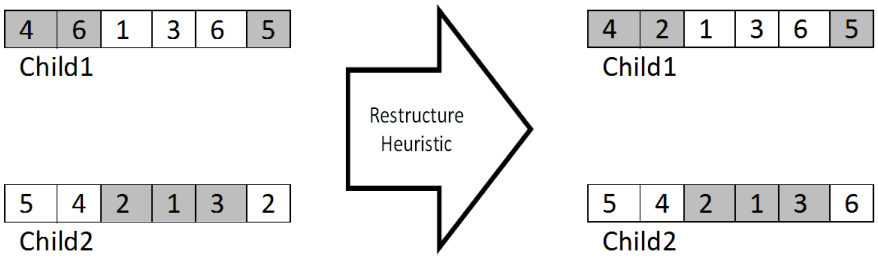

Figure 2. Restructure of offsprings

3.1.6. Implementation of Mutation Operator

The aim of mutation is to increase the variability in chromosomes of the population and to direct the population into an unexplored area of the search space. It often directs the execution to escape from local optima. If the population contains n chromosomes, m is the number of genes in each chromosome and the rate of mutation is P_m, then $n \times m \times P_m$ genes are randomly chosen for mutation. It generates new chromosome as demonstrated in Figure 3.

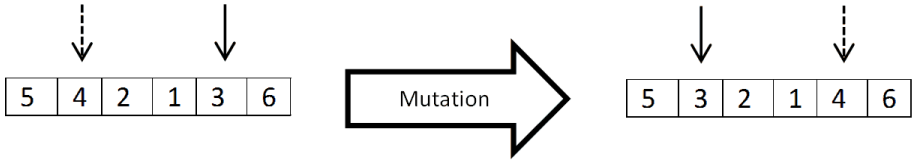

Figure 3. Mutation operation for a chromosome of (6×6) test problem

The procedure generates two random integers, *r1* and *r2* between 0 and *m*. Then swaps the $r1^{th}$ and $r2^{th}$ genes as shown in Figure 3. The procedure of the mutation operation is presented as,

```
for i=1:ceil(popSize*m*prob_Mutation)
    p=randint(1,1,[1,popSize]);
    b=Population(p,:);
    r1=randint(1,1,[1,len]);
    r2=randint(1,1,[1,len]);
    tmp=b(r1);
    b(r1)=b(r2);
    b(r2)=tmp;
    Population(p,:)=b;
  end
```

3.1.7. Implementation of Immune Operator

The procedure of forward immune operator is presented as:

```
for i=1:ceil(popSize*prob_Immune)
    e(i)=randint(1,1,[1,popSize]);
    tmp= Population(e(i),:);
    for i=1:m
        r1=randint(1,1,[1,m]);
        tmp1=tmp(r1);
        tmp(r1)=tmp(i);
        tmp(i)=tmp1;
        temporary_Fitness=f(tmp);
        if (temporary_Fitness<=Elitist_Fitness)
                Population(e(i),:)=tmp;
        else if (rand<=exp((Elitist_Fitness-temporary_Fitness)/(k*T)))
                Population(e(i),:)=tmp;
            end
        end
    end
end
```

We select a position within a chromosome at random and exchange the first position and the randomly selected position to produce a new offspring. Then, we perform the immunity test based on Boltzmann's acceptance probability of Simulated Annealing (SA). Here we set the value of *k=1* and *T= log (Elitist_Fitness /generation_count + 1)*. If the new offspring is good enough we include it into the population else the next to the first gene (right) of the offspring is altered in the same way until all the locations have been considered.

The backward immune operator works in the same way except that we select the last gene of the offspring in place of the first gene and proceeds to the left. The procedure is,

```
for i= ceil(popSize*prob_Immune):1
    e(i)=randint(1,1,[1,popSize]);
    tmp= Population(e(i),:);
    for i=1:m
        r1=randint(1,1,[1,m]);
        tmp1=tmp(r1);
        tmp(r1)=tmp(i);
        tmp(i)=tmp1;
        temporary_Fitness=f(tmp);
        if (temporary_Fitness<=Elitist_Fitness)
                Population(e(i),:)=tmp;
        else if (rand<=exp((Elitist_Fitness-temporary_Fitness)/(k*T)))
                Population(e(i),:)=tmp;
            end
        end
    end
end
```

If the population contains n (*popSize=n*) chromosomes and the immunity rate is P_v, then $n \times P_v$ chromosomes randomly chosen for immune operation. Whether to choose forward or backward immune operator that depends on a random number r_n, if $r_n <= 0.5$ we go for

forward operator else we go for backward operator. The forward and backward operations are depicted in Figure 4 for two iterations.

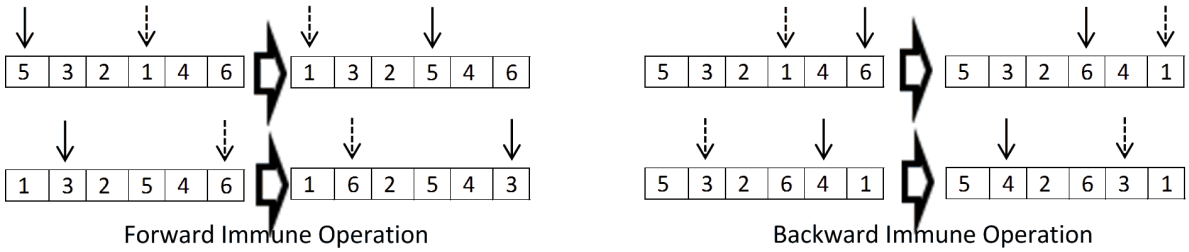

Figure 4. Immune operation for a chromosome of (6×6) test problem

3.1.8. Elitist Replacement Strategy

For every generation we specifically store the worst chromosome separately and we replace this with the elitist chromosome of that particular population. For an example in the i^{th} generation once we perform all the genetic operations on the population *P*, we obtain a new population *newP*. We evaluate each of the chromosomes of *newP* and store the best one as $elitist_i$. Then if p_{i+1} is the worst chromosome in $(i+1)^{th}$ generation then we store it to a variable called $worst_{i+1}$ and replace $worst_{i+1}$ with $elitist_i$. This phenomenon eventually improves the searching direction of the algorithm since it follows the schema of the good solutions.

3.1.9. Stopping Condition

It governs the execution of the Immune-GA-RS algorithm. It implies that all the operators are executed repeatedly until a stopping condition is encountered. The execution of IGA is eventually terminated if it doesn't observe any improvement for a long interval of execution or if it reaches the maximum number of generations count. The flowchart of Immune-GA-RS is depicted in Figure 5.

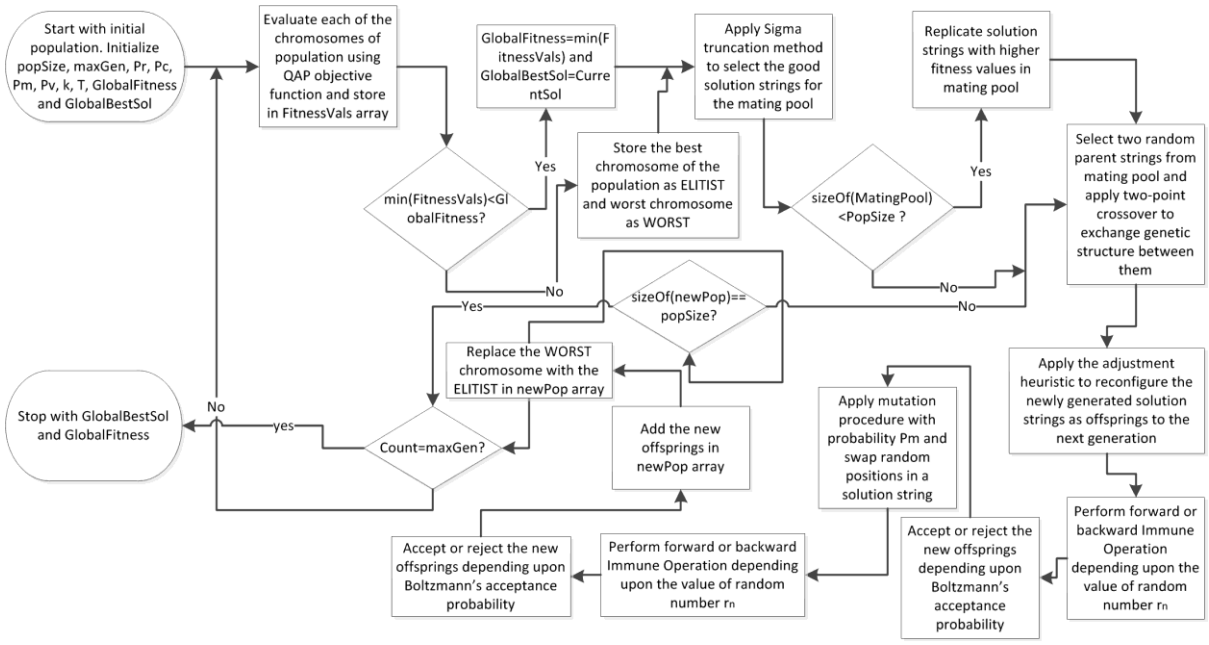

Figure 5. Flow Chart of Immune-GA-RS

4. Experimental Verifications

The proposed Immune-GA-RS algorithm is coded in Matlab 7.6 using an Intel PC with Quad CPU (2.83 GHz) and 4 GB of RAM. The computational complexity of the Immune-GA-RS intensifies exponentially with the size of the problems. Therefore a good design is indeed important while dealing with the large test problems. The evaluation criterion of Immune-GA-RS is based on total inter-cell material flow cost. To determine the desired values of the parameters of Immune-GA-RS, extensive experiments are accomplished. Parameters of Immune-GA-RS, population size (*PopSize*), number of maximum generations (*MaxGen*), probability of reproduction (P_r), probability of crossover (P_c) and probability of mutation (P_m) and probability of vaccination (immunity) (*Pv*) are to be fixed to obtain good results. The values of the parameters are reported in Table 2.

Table 2. Parameter values for the Immune-GA-RS

No. of cells	*PopSize*	*MaxGen*	P_r	P_c	P_m	*Pv*
<=25	100	2000	0.6	0.5	0.05	0.3
>25	100	4000	0.6	0.5	0.05	0.3

These values are obtained after extensive analysis with different settings of parameters on all the datasets. The decision of keeping the low CPU time is a crucial trade-off in the context of the QAP since the Immune-GA-RS is intended to get trapped in local optima with reduced values of parameters. In order to show the competence of the proposed Immune-GA-RS first we compared our results with the published results. For the comparison purpose we have developed two more variants of genetic algorithms namely GA-RS (genetic algorithm with replacement strategy) and ALT-GA-RS (alternative genetic algorithm with replacement strategy). The flowcharts of these algorithms are depicted in appendix. From the open CLP literature we have understood that the test problems suggested by Nugent, Vollman and Ruml (1968) are being widely used by the CLP researchers to verify their techniques. Thus we have used the same datasets for the experimental verifications. To establish the proposed Immune-GA-RS we have compared our results with all the available CLP methodologies such as Bubble Search (Wang and Sarker, 2002), ACO (Solimanpur, Vrat and Shankar, 2004), HGA (Singh, 2007) and GA (Kulkarni and Shankar, 2007) along with the popular layout design techniques such as HC63, HC63-66 and CRAFT (Nugent, Vollman, and Ruml, 1968). Table 3 demonstrates the results obtained by the proposed Immune-GA-RS and also the published results.

Table 3. Comparison of Immune-GA-RS with other published techniques

No. of Cells	H63	HC63-66	CRAFT	Bubble Search	HGA	K&K GA	ACO	GA-RS	Alt-GA-RS	Immune-GA-RS	Time
Nug5	27.6	29.4	28.2	25.2	**25**	**25**	**25**	**25**	**25**	**25**	0.01
Nug6	44.2	44.2	44.2	43	**43**	**43**	**43**	**43**	**43**	**43**	0.01
Nug7	78.8	78.4	78.4	75	**74**	**74**	**74**	**74**	**74**	**74**	0.12
Nug8	114.4	110.2	113.4	109	116	**107**	**107**	**107**	**107**	**107**	0.26
Nug12	317.4	310.2	296.2	301.6	296	302	**289**	300	**289**	**289**	2.3
Nug15	632.6	600.2	606	585.8	592	640	**575**	601	**575**	**575**	6.82
Nug20	1400.4	1345	1339	1332.6	1314	1474	**1285**	1310	1287	**1285**	25.12
Nug30	3267.2	3206.8	3189	3165.2	3482	3518	3062.4	3128	3097	**3062**	74.4

All these test problems (Nug5-Nug30) are available in *QAPLIB-A Quadratic Assignment Problem Library* (http://www.seas.upenn.edu/qaplib/). The global optimal solutions are also available in that stated library. We have executed all the three methods 10 times for each and every problem tested and picked the best objective value achieved. Table 3 clearly states that Immune-GA-RS is capable enough of achieving the best results which are optimal according to the QAPLIB. It outperforms all the other 9 techniques in terms of solution quality. We have also recorded the computational time of Immune-GA-RS as provided in last column of Table 3. The average computing time calculated is 13.63 CPU Seconds. Only Solimanpur, Vrat and Shankar (2004) demonstrated the CPU Seconds of the ACO method for all these problems. The average of the CPU time of their method is 14.77. Thus we established a better methodology which is 11.11% improved in terms of the quality of solutions and 7.72% improved in terms of computational time. Figure 6 depicts the pictorial view of the improved performances of Immune-GA-RS.

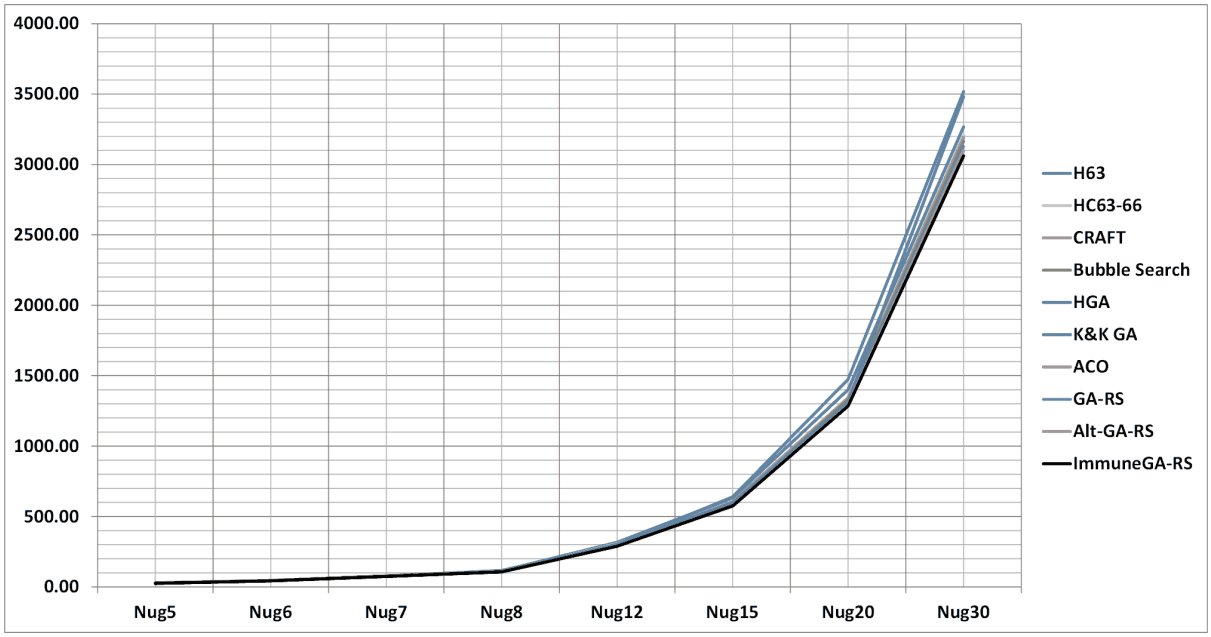

Figure 6. Performance curve of Immune-GA-RS and other methods for Nug5-Nug30

In the next stage of our experiment we tested the proposed Immune-GA-RS with various dimensions of the QAP instances obtained from QAPLIB. Solimanpur, Vrat and Shankar (2004) mentioned that it is unnecessary to consider problems larger than $N=30$ since in real scenario more than 30 cells are rarely used in the factory shop-floor. Yet we have tested our methodology for even one of the largest datasets available in QAPLIB (wil100) in order to realize the ability of Immune-GA-RS while solving the tangibly large QAP. The results are reported in Table 4. According to the QAPLIB there is hardly any technique available in past literature which can provide optimal solutions for all the different QAP instances singly. We have tested total 36 structured QAP instances for which $12 \leq N \leq 100$. Along with Immune-GA-RS other two GA variants are also tested for the performance comparisons. All the methodologies are compared in terms of the objective values achieved and the computational time consumed. Table 4 depicts that Immune-GA-RS is extremely efficient in terms of the obtained solution quality. It attains global optimal solutions for 33 test instances while near-optimal solutions for the remaining 3 problems. From the very last column of Table 4 we decode that on an average Immune-GA-RS took 28.42 CPU Seconds for all the 36 structured QAP instances. Thus Immune-GA-RS outperforms the other two variants of GA in terms of CPU time also (Figure 7). Thus we achieved 91.67% improved solutions while compared to

OS/BSF and GA variants. Furthermore we compared the results of Immune-GA-RS with recently published five QAP solution methodologies, SA (Singh and Sharma, 2008), GA (Singh and Sharma, 2010), SC-TS (Fescioglu-Unvera and Kokar, 2011), IHGA (Misevicius, 2004), GA-(Kratica et al., 2011) respectively. Table 5 demonstrates the outcome of this comparison. It shows that all the 36 QAP instances have rarely been considered collectively in any recent article. However these five methodologies are contemporary and capable of producing improved solutions for QAP while compared to past literature. The proposed Immune-GA-RS is shown to attain smaller solution gap for 11 instances and outperforms all the five published algorithms. Conversely for 1 problem instance (Tai25b) Immune-GA-RS achieved 0.09% deviation from the optimal solution of IHGA (Misevicius, 2004). There are 5 test instances (lipa20a, lipa20b, Nug25, Lipa30b and Lipa50b) for which we could not compare the results obtained by Immune-GA-RS since none of those five algorithms is used to solve those test problems. However Immune-GA-RS achieved 'zero' solution gap for those 5 instances, therefore we do not need any comparison for that matter.

Table 4. Comparison of results obtained by Immune-GA-RS, ALT-GA-RS and GA-RS for the standard QAP instances obtained from QAPLIB and their optimal solutions

Instances	OS/BSF*	GA-RS	CPU time	ALT-GA-RS	CPU time	Immune-GA-RS	CPU time
Chr12a	**9552**	9938	6.10	**9562**	5.10	**9552**	1.20
Chr12b	**9742**	9762	9.60	**9742**	8.90	**9742**	1.50
Chr12c	**11156**	11834	15.40	11662	6.50	**11156**	3.70
Had12	**1652**	1660	20.09	**1652**	10.90	**1652**	28.10
Rou12	**235528**	240124	55.90	241700	9.30	**235528**	0.31
Scr12	**31410**	32236	7.44	**31410**	3.90	**31410**	25.50
Tai12a	**224416**	238860	5.65	**224416**	26.10	**224416**	1.20
Tai12b	**39464925**	40376142	13.30	**39464925**	6.50	**39464925**	2.80
Chr15a	**9896**	11372	6.70	10122	12.30	**9896**	11.20
Chr15b	**7990**	10166	2.30	8640	34.90	**7990**	2.50
Chr15c	**9504**	11488	10.20	**9504**	19.60	**9504**	3.40
Rou15	**354210**	368446	19.80	**354210**	8.40	**354210**	21.30
Scr15	**51140**	53182	8.80	54144	12.90	**51140**	16.10
Tai15a	**388214**	398394	15.60	388402	31.20	**388214**	1.50
Tai15b	**51765268**	52002324	2.92	**51765268**	8.40	**51765268**	53.11
Chr20a	**2192**	2726	13.35	2290	31.20	**2192**	48.10
Chr20b	**2298**	3006	19.85	2576	12.16	**2298**	10.60
Chr20c	**14142**	18658	15.43	17584	21.32	**14142**	45.90
Had20	**6922**	6940	9.20	**6922**	25.70	**6922**	1.90
Lipa20a	**3683**	3767	29.50	**3683**	17.20	**3683**	9.80
Lipa20b	**27076**	30960	24.30	**27076**	21.20	**27076**	16.10
Rou20	**725522**	734156	7.70	727322	11.90	**725522**	6.10
Scr20	**110030**	113460	9.36	**110030**	12.10	**110030**	2.10
Tai20a	**703482**	720830	4.80	**703482**	12.40	**703482**	32.75
Tai20b	**122455319**	124358060	108.42	123062505	7.70	**122455319**	10.20
Nug25	**3744**	3912	19.90	3790	52.10	**3744**	21.45
Tai25a	**1167256**	1217670	72.41	1216392	55.50	**1167256**	12.88
Tai25b	**344355646**	364570560	27.30	353995465	48.40	344390003	16.70
Kra30a	**88900**	96250	19.20	95260	25.80	**88900**	7.30
Kra30b	**91420**	94400	22.60	93350	27.21	**91420**	12.10
Lipa30a	**13178**	13477	19.30	**13178**	47.20	**13178**	7.70
Lipa30b	**151426**	175979	42.10	160066	11.90	**151426**	44.60
Lipa50a	**62093**	63107	161.20	63070	98.11	**62093**	101.80
Lipa50b	**1210244**	1442621	128.20	1336731	113.30	**1210244**	84.70
Wil50	**48816**	50030	89.90	49916	78.43	48851	84.20
Wil100	**273038**	284704	418.45	284736	336.60	273078	272.62

* *OS/BSF: optimal solution/best solution found*

To further prove the competence of the Immune-GA-RS an elaborated statistical analysis is performed on the results obtained for all the 44 problems (36 QAPs + 8 CLPs). The details

are given in Table 6 and Table 7. First we have performed the F-test two samples for variances among the two set of data obtained for OS/BSF (from QAPLIB) and Immune-GA-RS for all the 44 instances. If test statistic < critical value ($F < F_{critical}$), we accept the null hypothesis. In other words if *p-value* > α, we accept the null hypothesis. Table 6 depicts $F < F_{critical}$ (*1.000174448 < 1.660743744*) and *p-value* > α (*0.499773161 > 0.05*), thus we accept the null hypothesis that the variances are equal. Next we have conducted the paired t-test assuming equal variances.

Table 5. Comparison of solution gap obtained by Immune-GA-RS and SA (S&S-2008), GA (S&S-2010), SC-TS (F&M-2011), IHGA (Mis-2004), GA-(KTF&D-2011)

Instances	OS/BSF*	SA (S&S-2008)	GA (S&S-2010)	SC-TS (F&M-2011)	IHGA (Mis-2004)	GA-(KTF&D-2011)	Immune-GA-RS
Chr12a	9552	0	-	-	-	0	0
Chr12b	9742	0	-	-	-	0	0
Chr12c	11156	0.26	-	-	-	0	0
Had12	1652	0	-	-	-	0	0
Rou12	235528	0	-	-	-	-	0
Scr12	31410	0	-	-	-	-	0
Tai12a	224416	0	-	-	-	-	0
Tai12b	39464925	0.3	-	-	-	-	0
Chr15a	9896	0	-	-	-	0.061	0
Chr15b	7990	2.7	-	-	-	0	0
Chr15c	9504	11.5	-	-	-	1.3	0
Rou15	354210	0.71	-	-	-	-	0
Scr15	51140	0	-	-	-	-	0
Tai15a	388214	0.39	-	-	-	-	0
Tai15b	51765268	0.47	-	-	-	-	0
Chr20a	2192	0	-	-	-	0.77	0
Chr20b	2298	0	-	-	-	5.01	0
Chr20c	14142	0	-	-	-	0.473	0
Had20	6922	0	1.5	-	-	0	0
Lipa20a	3683	-	-	-	-	-	0
Lipa20b	27076	-	-	-	-	-	0
Rou20	725522	0.06	-	-	-	-	0
Scr20	110030	2.13	2.3	-	-	-	0
Tai20a	703482	0.21	3.6	0.246	0	-	0
Tai20b	122455319	5.6	2.3	-	0	-	0
Nug25	3744	-	-	-	-	-	0
Tai25a	1167256	-	-	0.239	0	-	0
Tai25b	344355646	-	-	0.702	0	-	0.009
Kra30a	88900	-	-	0.714	0	-	0
Kra30b	91420	-	-	0.178	0	-	0
Lipa30a	13178	-	3.2	-	-	-	0
Lipa30b	151426	-	-	-	-	-	0
Lipa50a	62093	-	2.4	-	-	-	0
Lipa50b	1210244	-	-	-	-	-	0
Wil50	48816	-	4	0.105	-	-	0.072
Wil100	273038	-	6.36	0.099	-	-	0.015

* *OS/BSF: optimal solution/best solution found*

The result of paired t-test is reported in Table 7. We interpret the two-sample t-test result as, if test statistic < critical value ($t < t_{critical}$), we accept the null hypothesis or in other word if *p-value* > α, we accept the null hypothesis. Since the null hypothesis is that the mean difference = 0, is a two-sided test. Therefore, we use both the one-tail and two-tail values for the analysis. Since the *t-statistic* < $t_{critical}$ (6.66022E-05 < 1.662765449) and 6.66022E-05 < 1.987934206) and both the *p-values* 0.499973507 (one tail) and 0.999947013 (two-tail) > α (α = 0.05), we accept the null hypothesis and state that the means are same i.e. the data are consistent. More specifically we can state that the Immune-GA-RS (mean = 12821157) is an

improved method and it is capable of obtaining similar quality solutions for the structured QAP problems of QAPLIB (mean of OS/BSF = 12820375) at 95% confidence level. Therefore we prove that Immune-GA-RS is an efficient layout designing technique.

Table 6. Two-Sample F-Test for variances of results obtained for Immune-GA-RS and OS/BSF (α =0.05)

	Immune-GA-RS	***OS/BSF***
Mean	12821157	12820375
Variance	3.03741E+15	3.03688E+15
Observations	44	44
df	43	43
F	1.000174448	
P(F<=f) one-tail	0.499773161	
F Critical one-tail	1.660743744	

4.1. Convergence Characteristics of Immune-GA-RS

Figure 8 displays the convergence property of Immune-GA-RS along with the other two variants of GA. These are almost equivalent for all the problem datasets. For explanation purpose lipa20b problem is selected. Convergence property is demonstrated during the iterations of the proposed techniques. Immune-GA-RS obtains best layout configuration along with ALT-GA-RS which is way better than the GA-RS technique. However Immune-GA-RS converges faster than ALT-GA-RS. Immune-GA-RS consumes around 700 iterations while ALT-GA-RS takes nearly 900 iterations to attain optimality. Due to the diversification property of immune operator Immune-GA-RS reaches to the more prominent area of search space faster. Thus we can conclude that the competency of Immune-GA-RS to escape from local optimal solution is better than other GAs since it constantly improves the solution till the end of the execution. All the three techniques confirm the same pattern of convergence characteristics for all the tested problems therefore the convergence property is established.

Table 7. Two-sample T-test assuming equal variance and equal sample size of results obtained for Immune-GA-RS and OS/BSF (α=0.05)

	Immune-GA-RS	***OS/BSF***
Mean	12821157	12820375
Variance	3.03741E+15	3.03688E+15
Observations	44	44
Pooled Variance	3.03714E+15	
Hypothesized Mean Difference	0	
df	86	
t-Statistic	6.66022E-05	
P(T<=t) one-tail	0.499973507	
$t_{Critical}$ one-tail	1.662765449	
P(T<=t) two-tail	0.999947013	
$t_{Critical}$ two-tail	1.987934206	

5. Conclusions

We have proposed a novel CLP methodology based on Immune-GA with elitist replacement strategy, namely Immune-GA-RS. We have adopted a QAP model of CLP proposed by Wang and Sarker (2002). QAP is considered as the hardest among all the optimization

problems (NP-Hard). Therefore a good design of the heuristic method is essential in this context. We have incorporated an elitist replacement strategy in the proposed technique which improves the searching efficiency of standard Immune-GA. We have also developed two more variants of GA namely, GA-RS and ALT-GA-RS for the testing purpose. For the verification of the proposed Immune-GA-RS, we have adopted 8 benchmark problems which are being profoundly used in CLP literature. We have compared Immune-GA-RS with total of 9 algorithms from published literature including the two variants of GA we have coded. Immune-GA-RS is shown to perform extremely well for all the datasets and obtains 11.11% improvement in terms of solution quality and 7.72% improvement in terms of CPU time.

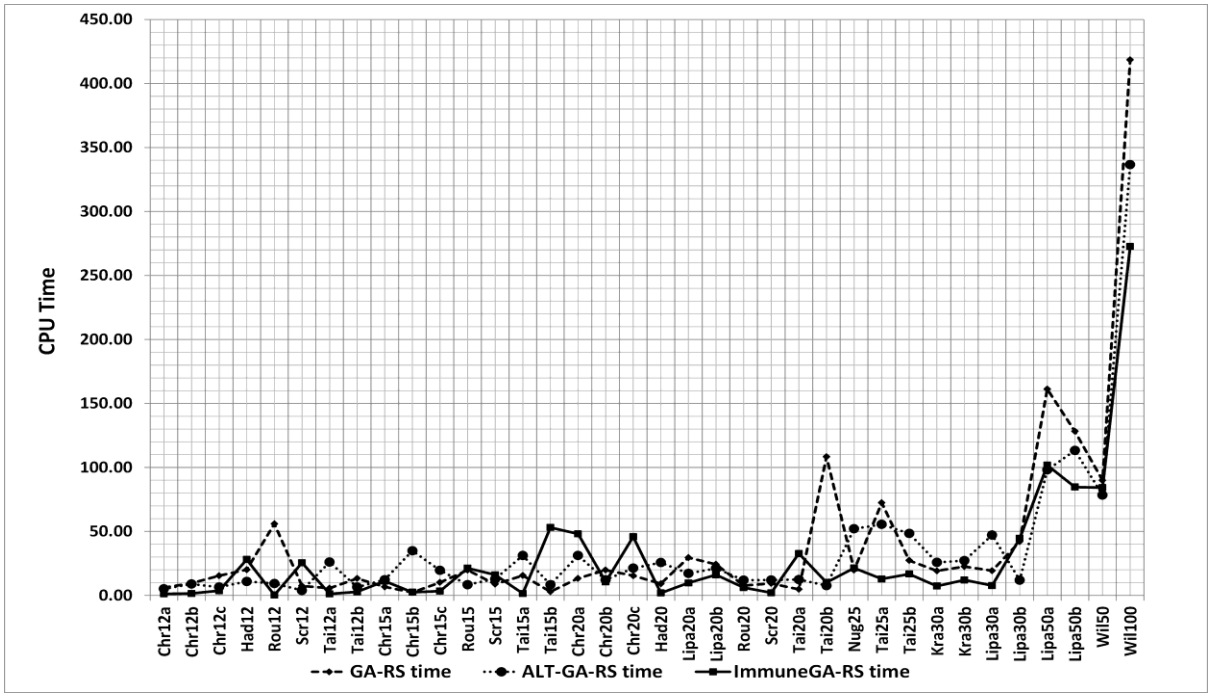

Figure 7. The comparison curve of Immune-GA-RS with other GA variants in terms of computational time for 36 systematized QAP instances of QAPLIB

In order to realize the efficiency of Immune-GA-RS, we have selected 36 structured QAP instances from QAPLIB where $12 \leq N \leq 100$. We have tested all the three techniques for these test problems and compared with the best known solutions of these problems. Immune-GA-RS is shown to achieve equally good solutions for 33 problems and near-best solutions for remaining 3 problems. Thus we achieved 91.67% improved solutions while compared to OS/BSF and have shown to outperform the recent QAP based algorithms, SA (S&S-2008), GA (S&S-2010), SC-TS (F&M-2011), IHGA (Mis-2004), GA-(KTF&D-2011), respectively with smaller solution gap for 11 instances and obtain equal or better solutions for 24 instances. It is also shown outperforms other two GA based methods not only in terms of solution quality but also for the CPU time (Figure 7). Further the statistical tests (F-test and paired t-test) are carried out to prove the competence of the proposed Immune-GA-RS. It depicts that the achieved solutions of Immune-GA-RS are as good as the best known solutions found in the literature. The novelty of our paper is twofold, (a) to foster a state-of-the-art meta-heuristic technique which not only outperforms all the standing CLP techniques available in the literature but also produces best solutions for the larger QAPs of QAPLIB while compared with recent self-controlling Tabu Search method, (b) to carry out statistical verification of solutions of the proposed technique which has never been carried out in the history of inter-cell layout methodologies.

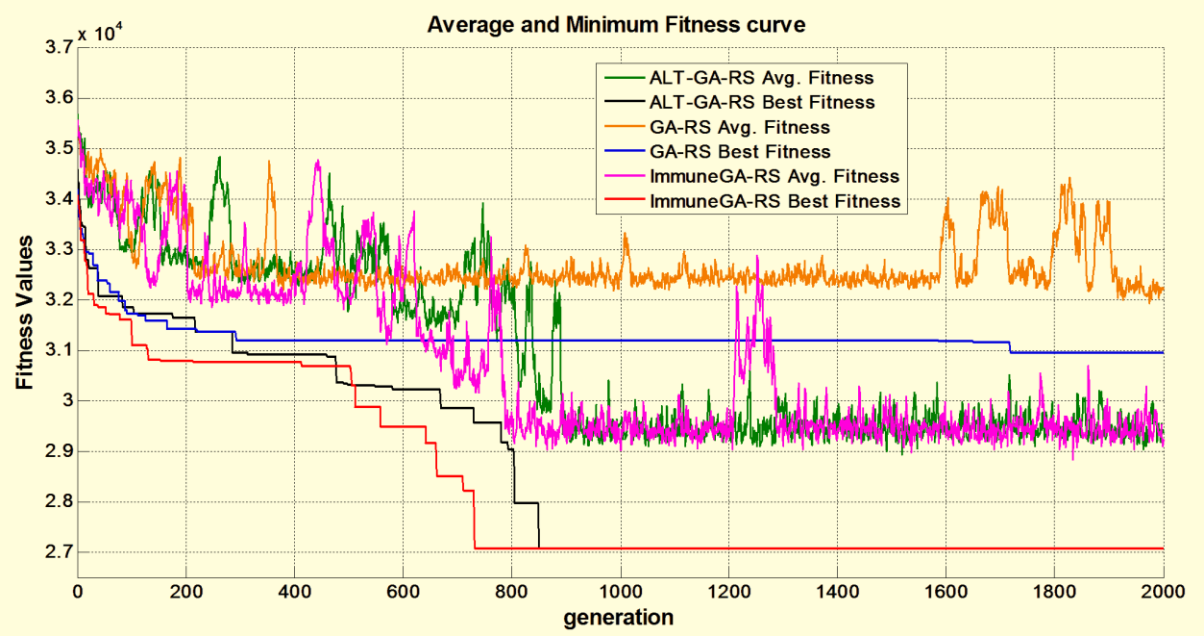

Figure 8. Sample convergence analysis for Immune-GA-RS and other two GA variants for lipa20b problem instance

Appendix

The flowcharts of GA-RS and Alt-GA-RS are displayed in Figure A1 and A2 respectively.

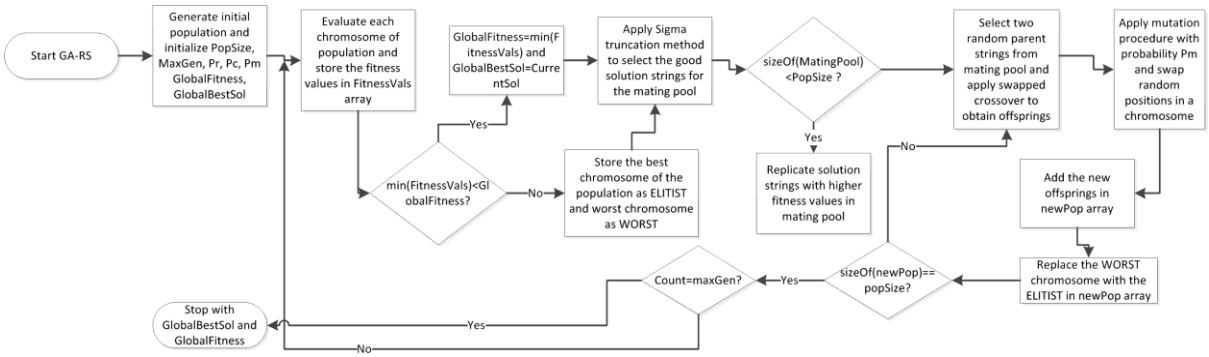

Figure A1. Flowchart of GA-RS

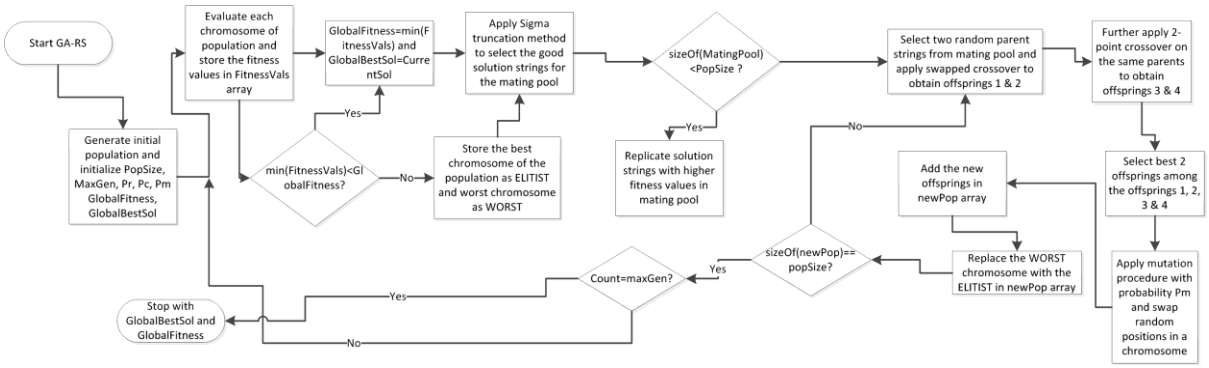

Figure A2. Flowchart of ALT-GA-RS

References

Ahi, A., Aryanezhad, M.B., Ashtiani, B., & Makui, A. (2009). A novel approach to determine cell formation, intracellular machine layout and cell layout in the CMS problem based on TOPSIS method. *Computers & Operations Research, 36(5)*, 1478–1496.

Ahuja, R. K., Orlin, J.B., & Tiwari, A. (2000). A greedy genetic algorithm for the quadratic assignment problem. *Computers and Operation Research, 27(10)*, 917–934.

Alfa, A.S. Chen, M., & Heragu, S.S. (1992). Integrating the grouping and layout problems in cellular manufacturing systems. *Computers & Industrial Engineering, 23(1-4)*, 55–58.

Anderson E.J., & Ferris, M.C. (1994). Genetic Algorithms for Combinatorial Optimization: The Assemble Line Balancing Problem. *INFORMS Journal on Computing, 6(2)*, 161-173.

Ariafar, S., & Ismail, N. (2009). An improved algorithm for layout design in cellular manufacturing systems. *Journal of Manufacturing Systems, 28(4)*, 132-139.

Arkat, J., Farahani, M.H., & Hosseini, L. (2011). Integrating cell formation with cellular layout and operations scheduling. *International Journal of Advanced Manufacturing Technology,* DOI 10.1007/s00170-011-3733-4.

Balakrishnan, J., & Cheng C.H. (2009). The dynamic plant layout problem: Incorporating rolling horizons and forecast uncertainty. *Omega, 37(1)*, 165-177.

Chan, F.T.S., Lau, K.W., Chan, P.L.Y., & Choy, K.L. (2006). Two-stage approach for machine-part grouping and cell layout problems. *Robotics and Computer-Integrated Manufacturing, 22(3)*, 217–238.

Chan, W.M., Chan, C.Y., & Kwong, C.K. (2004). Development of the MAIN algorithm for a cellular manufacturing machine layout. *International Journal of Production Research, 42(1)*, 51–65.

Chen, R. (1980). *The Bases of Immunology*. Peking: Health Press.

Darwin, A. (1929). *The Origin of Species by Means of Natural Selection or the Preservation of Favored Races in the Struggle for Life*. The Book League of America (originally published in 1859)..

Drezner, Z. (2003). A New Genetic Algorithm for the Quadratic Assignment Problem. *INFORMS Journal on Computing, 15(3)*, 320–330.

Fescioglu-Unvera, N., & Kokar, M.M. (2011). Self-Controlling Tabu Search algorithm for the Quadratic Assignment Problem. Computers & Industrial Engineering, *60(2)*, 310–319.

Fisher, R.A. (1930). *The genetical theory of natural selection*. Oxford: Clarendon Press.

Fleurent, C., & Ferland, J.A. (1994). Genetic hybrids for the quadratic assignment problem. In Pardalos, P., & Wolkowicz, H. (Eds.). *Quadratic Assignment and Related Problems*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, *16*, 173–187.

Gen, M., & Cheng, R. (2000). *Genetic Algorithms and Engineering Optimization*. John Wiley & Sons.

Ghosh, T., Dan, P.K., Sengupta, S., & Chattopadhyay, M., (2010). Genetic rule based techniques in cellular manufacturing (1992-2010): a systematic survey, *International Journal of Engineering, Science and Technology*, 2(5), 198-215.

Goldberg, D.E. (1989). *Genetic Algorithms in Search Optimization & Machine Learning*. Addison Wesley.

Heragu, S.S., & Kakuturi, S.R. (1997). Grouping and placement of machine cells. *IIE Transactions, 29(7)*, 561-571.

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI.

Ji, P., Wu, Y., & Liu, H. (2006). A Solution Method for the Quadratic Assignment Problem (QAP). *The Sixth International Symposium on Operations Research and Its Applications (ISORA, 2006)*, Xinjiang, China, 106-117.

Jiao, L., & Wang, L. (2000). A Novel Genetic Algorithm Based on Immunity. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans, 30(5)*, 552-561.

Jolai, F., Taghipour, M., & Javadi, B. (2011). A variable neighborhood binary particle swarm algorithm for cell layout problem. *International Journal of Advanced Manufacturing Technology*, 55(1-4), 327–339.

Kaufman, L., & Broeckx, F. (1978). An Algorithm for the Quadratic Assignment Problem Using Benders Decomposition. *European Journal of Operational Research, 2(3),* 207-211.

Kia, R., Baboli, A., Javadian, N. Tavakkoli-Moghaddam, R., Kazemi, M., & Khorrami, J. (2012). Solving a group layout design model of a dynamic cellular manufacturing system with alternative process routings, lot splitting and flexible reconfiguration by simulated annealing. *Computers & Operations Research, 39(11),* 2642–2658.

Kratica, J., Tosic, D., Filipovic, V., & Dugosija, D. (2011). A New Genetic Representation for Quadratic Assignment Problem. *Yugoslav Journal of Operations Research, 21(2),* 225-238.

Kulkarni, P.C., & Shanker, K. (2007). A Genetic Algorithm for Layout problems in Cellular Manufacturing Systems. *Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management*, 694-698.

Lee, S.D. Configuring layout for a cellular manufacturing system. *International Journal of Systems Science, 29(6),* 557–564.

Leno, I.J., Sankar, S.S., Raj, M.V., & Ponnambalam, S.G. (2011). Bi-criteria Optimization in Integrated Layout Design of Cellular Manufacturing Systems Using a Genetic Algorithm. *Swarm, Evolutionary, And Memetic Computing, Lecture Notes in Computer Science, 7076*, 323-331.

Logendran, R. (1991). Impact of sequence of operations and layout of cells in cellular manufacturing. *International Journal of Production Research, 29(2),* 375-390.

Ma, H., & Zhang, D. (2010). The Dynamics Facility Layout Study Based On Cellular Manufacturing. *2010 IEEE International Conference on Measuring Technology and Mechatronics Automation*, Zhenjiang, China, 3, 862-865.

Mahdavi, I., & Mahadevan, B. (2008). CLASS: An algorithm for cellular manufacturing system and layout design using sequence data. *Robotics and Computer-Integrated Manufacturing, 24(3),* 488–497.

Misevicius, A. (2004). An improved hybrid genetic algorithm: new results for the quadratic assignment problem. *Knowledge-Based Systems, 17*, 65–73.

Mitchell, M. (1999). *An Introduction to Genetic Algorithms*. A Bradford Book The MIT Press, Cambridge, MA.

Nugent, C.E., Vollman, T.E., & Ruml, J., (1968). An experimental comparison of techniques for the assignment of facilities to locations. *Operations Research, 16(1),* 150–173.

Sarker, B.R., & Yu, J.A. (1994). Two-phase procedure for duplicating bottleneck machines in a linear layout, cellular manufacturing system. *International Journal of Production Research, 32(9),* 2049-2067.

Salum, L. (2000). The Cellular manufacturing layout problem. *International Journal of Production Research, 38(5),* 1053-1069.

Selim, M.S., Askin, R.G., & Vakharia, A.J. (1998). Cell formation in group technology: review evaluation and directions for future research. *Computers & Industrial Engineering, 34(1),* 3–20.

Singh, S.P. (2007). *Solving single and multi-period facility layout problems: few SA and GA based heuristics.* Unpublished Ph.D. Thesis, Indian Institute of Technology, Kanpur, India.

Singh, S. P., & Sharma, R. R. K. (2008). Two-level modified simulated annealing based approach for solving facility layout problem. *International Journal of Production Research, 46(13),* 3563-3582.

Singh, S. P., & Sharma, R. R. K. (2010). A Hybrid Genetic Search Based Approach to Solve Single Period Facility Layout Problem. *Asia Pacific Management Review, 15(2),* 301-312.

Solimanpur, M., Vrat, P., & Shankar, R. (2004). Ant colony optimization algorithm to the inter-cell layout problem in cellular manufacturing. *European Journal of Operational Research, 157(3),* 592–606.

Tang, C., & Abdel-Malek, L.L. (1996). A framework for hierarchical interactive generation of cellular layout. *International Journal of Production Research, 34(8),* 2133-2163.

Tate, D.M., & Smith, A.E. (1995). A genetic approach to the quadratic assignment problem. *Computers and Operation Research, 22(1),* 73–83.

Tavakkoli-Moghaddam, R., Javadian, N., Javadi, B., & Safaei, N. (2007). Design of a facility layout problem in cellular manufacturing systems with stochastic demands. *Applied Mathematics and Computation, 184(2),* 721–728.

Wang, S., & Sarker, B.R. (2002). Locating Cells with Bottleneck Machines in Cellular Manufacturing Systems. *International Journal of Production Research, 40(2),* 403-424.

Won, Y., & Currie, K.R. (2007). Fuzzy ART/RRR-RSS: a two-phase neural network algorithm for part-machine grouping in cellular manufacturing. *International Journal of Production Research, 45(9),* 2073–2104.

Wu, X., Chu, C.-H., Wang, Y., & Yan, W. (2006). Concurrent design of cellular manufacturing systems: a genetic algorithm approach. *International Journal of Production Research, 44(6),* 1217–1241.

Wu, Y., & Ji, P., (2007). Solving the Quadratic Assignment Problems by a Genetic Algorithm with a New Replacement Strategy. *World Academy of Science, Engineering and Technology, 30,* 310-314.

***Highlights (for review)**

- We have proposed a novel Immune Genetic Algorithm (Immune-GA-RS) to obtain competent inter-cell layout in the vicinity of CMS.
- The proposed method is successfully tested with 8 datasets which are being widely used for inter-cell layout design problems.
- Immune-GA-RS is compared with two variants of the Genetic Algorithms, GA-RS and Alt-GA-RS and 7 other published layout design techniques. Results portray that Immune-GA-RS acquires 11.11% improved solutions with 7.72% reduced CPU time on an average.
- Immune-GA-RS is tested on 36 structured QAP instances available through QAPLIB and shown to outperform other two GA variants while attaining optimal solutions for 33 instances and also shown to outpace five recent QAP based algorithms while attaining smaller solution gap for 11 test instances and obtain at least equal or better quality solutions for 24 instances.