Dynamic machine layout for press tool operations using real coded genetic algorithm

Kailas V. Chandratre*, Padmakar J. Pawar and Keshav N. Nandurkar

Production Engineering Department,
Savitribai Phule Pune University,
K.K Wagh Institute of Engineering Education and Research,
Panchawati,
Nashik 422003,
Pune, Maharashtra, India
Email: ckailas@rediffmail.com
Email: pjpawar1@rediffmail.com
Email: keshav1965@gmail.com
*Corresponding author

Abstract: In today's economy, manufacturing plants must be able to operate efficiently and respond quickly to changes in the product mix and demand. Layout design has a significant impact on manufacturing efficiency. A static plant layout if possible to be converted to dynamic layout may improve the efficiency of the plant significantly. Dynamic layout is a layout, which can be rearranged with respect to time as per variation in product design, quantity and change in product mix. Dynamic layout problem is a quadratic assignment problem and is of non-deterministic polynomial-time hard problem. In this work, an attempt is made to solve this problem using real coded genetic algorithm (GA), which overcomes some of the limitations of traditional GA. This algorithm has been applied to the dynamic layout benchmark problems to prove its effectiveness. In addition, a real life example is considered to validate the presented approach.

Keywords: dynamic layout; press working; real coded genetic algorithm.

Reference to this paper should be made as follows: Chandratre, K.V., Pawar, P.J. and Nandurkar, K.N. (2016) 'Dynamic machine layout for press tool operations using real coded genetic algorithm', *Int. J. Metaheuristics*, Vol. 5, No. 2, pp.91–114.

Biographical notes: K.V. Chandratre received his BE from Pune University, his ME from Shivaji University and is a research scholar in Production Engineering Department at K.K Wagh Institute of Engineering Education and Research, Nashik. He has 18 years of teaching experience and is currently working as an Associate Professor in the Mechanical Engineering Department of G.E.S.'s R.H. Sapat College of Engineering Management Studies and Research, Nasik. His research interests include advanced manufacturing processes, optimisation techniques.

P.J. Pawar received his BE from Pune University, his ME from Shivaji University and PhD from SVNIT Surat, India. He has 17 years of teaching and research experience and is currently working as a Professor in the Production

Engineering Department of K.K.W.I.E.E. & R, Nashik. His research interests include advanced manufacturing processes and optimisation techniques.

K.N. Nandurkar received his BE from Shivaji University, his ME from Mumbai University and PhD from IIT, Bombay. He has 26 years of teaching and research experience and is currently working as a Principal of K.K.W.I.E.E. & R, Nasik, and also a Professor in Production Engineering Department. His research interests include optimisation, industrial engineering, and cellular manufacturing.

1 Introduction

Layout design has invariably a significant impact on the performance of a manufacturing or service industry system, and consequently has been an active research area for several decades. Layout is the arrangement of facilities, machines, or departments in the plant, so that it is convenient for movement, operations of man, material and reduces the material handling. Generally, once the plant layout is finalised, it is not changed. This type of layout is called as static layout. However, in due course of time, because of change in product quantity or product mix or change in product itself, the static layout may not remain optimum. It may result in more material handling cost and more throughput time or delay. So, to overcome this problem, the layout is changed from time to time with respect to change in product mix or quantity. This layout is called as dynamic layout. In today's competitive era where there are frequent changes in the products, product mix and quantity, the static layout may not be effective as it will result into more material handling and subsequently higher material handling cost. Therefore, it is essential to consider the dynamic approach in such cases to increase the effectiveness of the plant.

Dynamic machine layout problem (DMLP) becomes more complex as the number of machines and the number of time periods is increasing. It is non-deterministic polynomial-time (NP) hard problem (Balakrishnan et al., 2003) and generally, this problem is formulated as quadratic assignment problem (QAP). Many times, solving the complex dynamic layout problem is difficult with traditional methods of optimisation (McKendall and Hakobyan, 2010). Hence, the researchers have been attempting metaheuristics such as genetic algorithm (GA), simulated annealing (SA), and tabu search (TS), and so on for solving dynamic layout problems. The dynamic layout problem has wide applications in press tool industries where the product mix is highly prone to change according to the market demand. Developments in the materials and machine handling technology are very useful for changing the machine layout frequently. Hence, in this work, dynamic layout problem for press tool industry is solved by applying one of the metaheuristics, known as real coded genetic algorithm (RCGA).

This paper is organised as follows. Section 2 provides a brief review of the literature on the dynamic layout problem. Section 3 discusses the formulation for DMLP and presented RCGA technique. In Section 4, the results of two benchmark problems are compared and a case study problem from industry is solved with presented RCGA. The conclusions are presented in Section 5.

2 Literature review

Traditionally, layout planning is treated as a strategic decision, which once implemented is difficult to modify. Advent of new material handling technology and materials used for manufacturing of modern machine tools has changed the perspective of layout planning. Development of mathematical models and use of recent and powerful problem-solving methodologies proved to be very useful to improve the effectiveness of plant operations. Various researchers have attempted to solve the dynamic layout problem using different approaches as discussed in this section.

Lenin et al. (2012) have presented a GA approach to find out machine's linear sequence with multi-products and different operation sequences for static layout. They have tried to minimise the total flow distance travelled by the product. They have generated some problems and their approach was compared with other approaches. Their approach was found to give more favorable results. However, they have not considered the dynamic conditions and only virtual problems are solved. Brunese and Tanchoco (2012) have studied single row machine layout problem with the within building constraints, because of which they could get improved quality of solutions in reduced time. To achieve these results, authors have used non-linear and mixed integer formulations. However, they have not considered the multi row layout and dynamic conditions. Bock and Hoberg (2007) have planned layout in more detailed way, by considering machine layout and transportation paths simultaneously. They have introduced new mathematical layout model and suggested several solution procedures to solve it. Promising results were obtained when it was tested with randomly generated instances. However, the authors have not considered the dynamic conditions and real life industry problems. Goncalves and Tiberti (2006) have proposed a new approach of GA, for the machine cell layout design. It is based on group encoding instead of simple machine encoding. Also new crossover and mutation operators are suggested. Their tests show that the algorithm is able to disclose the group structure. However, the authors have not considered the changing product mix and quantity. Solimanpur and Kamran (2010) have assumed a shop floor with multiple products, on multiple machines; and the alternate processing routes are considered to minimise total distance travelled by material. GA is applied to solve this problem. The effectiveness of the approach is evaluated by numerical example and results show that applied GA is effective and efficient. However, the applied approach does not consider the changing demand.

The aforementioned review shows that most of the researchers have considered the static nature of layout for optimisation. However, current scenario of industry shows the trend of changing demand, change in product mix and quantity. Therefore, to cope up with this changing scenario it is necessary to consider dynamic layout concept. Though dynamic layout is a new approach to industry, few researchers have worked on it.

Moslemipour and Lee (2012) have suggested QAP formulation to solve DMLP in flexible manufacturing system (FMS). They have considered the variation in product demand for changing the layout and have solved two randomly generated dynamic problems. These are solved with SA approach and have obtained satisfactory results. However, the problems they have considered are computer generated and virtual. Donga, Wu and Hou (2009) have studied dynamic multistage facility layout problem, in which new machines can be added or removed at different planning periods. They have proposed and used an auction algorithm, which is based on SA algorithm. To illustrate the proposed methodology, an industrial case of 4 periods and 56 machines is employed.

The results obtained are efficient. Azimi and Charmchi (2012) have suggested a new heuristic algorithm, which is developed by combining mathematical programming and simulation methods for dynamic facility layout problem with budget constraint. Proposed algorithm was tested over wide range of problems and the improved results were obtained. However, real case study needs to be solved. Mohammadi and Moghadam (2011) have studied dynamic machine layout for automotive shop and suggested a model to solve it. A problem of seven machines and seven-time periods is assumed, and satisfactory results are obtained. However, they have not considered the changing product mix. Ashtiani, Aryanezhad and Moghaddam (2007) have discussed the dynamic plant layout problem. They have proposed multi start SA approach to solve this problem. To compare the performance of the approach, some data sets are selected from the literature studies and resulted in good performance. Krishnan, Cheraghi and Nayak (2008) have developed 'Dynamic from between chart' approach to solve dynamic plant layout problem. A case study from Aircraft Company has been considered for verification of approach. However, they have not considered machines layout. Ahin and Turkbey (2008) have presented a new approach 'TABUSA' to solve the dynamic facility layout problem. The presented heuristic is based on SA approach supplemented with tabu list. The efficiency of the approach is tested against the data sets from literature studies. Baykasoglu and Gindy (2001) have used SA to solve dynamic layout problem. Some problems from literature studies are solved and the results are compared. Jinde and Telsang (2014) apply a GA approach to electrical motor manufacturing company to minimise material handling cost between machines. The problem is formulated in QAP and the results show that GA approach is performing well. However, the authors have considered static layout problem only.

The aforementioned literature review shows that most of the researchers have worked on design of dynamic plant layout, using various algorithms and various approaches. Still there is scope to solve DMLP by using other approaches. With various changing parameters, RCGA can also be seen as a new approach for solving these problems.

Maiti, Bhunia and Maiti (2006) have applied RCGA, in solving multiple price break structure and implemented for multi-item deterministic inventory control system having two separate storage facilities (owned and rented warehouse) due to limited capacity of the existing storage (owned warehouse). Their aim is to determine the optimal shipments, lot size of the two warehouses. Pal et al. (2005) have discussed an application of real-coded genetic algorithm (RCGA) for mixed integer non-linear programming in a two warehouses inventory control problem. The objective is to determine an optimal replenishment number, lot size of two warehouses. Wang Yu-Fen and Guo Xiao-Juan (2009) have applied RCGA in the automatic composing test paper system of the computer basis and claimed to have better performance than binary-coded GA. Rao et al. (2013) have applied RCGA to economic lot size scheduling problem (ELSP). Scheduling of production of different items over the same facility on a repetitive basis was the problem under consideration. Authors have claimed better results with RCGA than those obtained in literature studies. Norouzi et al. (2011) have applied integer coded genetic algorithm, to design the loading pattern (LP), in pressurised water reactors. Because of the large number of possible combinations for the fuel assemblies, loading in a core and finding the optimum solution is very complex. The authors have reported that application of a traditional ICGA to solve the linear programming/multiple integer linear programming (LP/MILP) problems somewhat suffers from long execution time and may not get the right (global optimum) solutions.

3 New approach proposed

When the flow of material between machines changes during a planning horizon, the static machine layout problem (SMLP) becomes dynamic, and this problem is known as the DMLP. The DMLP is based on the anticipated changes in flow that can occur in the future. The prospective future is divided into a number of time periods and the period may be defined in weeks, months, and so on. The solution for the SMLP is a single layout of machines, and the solution for the DMLP is a series of layouts, where each layout is associated with a particular period. This problem is of the category of NP hard problem. (McKendall and Hakobyan, 2010).

3.1 Mathematical formulation

The aforementioned problem is more commonly formulated as the QAP, and it is taken from Balakrishnan et al. (2003), where minimising total material handling cost Z is the objective function, it can be expressed in terms of the variables like *f, d,* and *t*.

$$\text{Min } Z = \sum_{t=1}^{P}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} f_{tik} d_{tjl} X_{tij} X_{tkl} + \sum_{t=2}^{P}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} A_{tijl} Y_{tijl} \tag{1}$$

Subject to

$$\sum_{i=1}^{n} X_{tij} = 1, \; j = 1\ldots.n, \; t = 1,\ldots., P \tag{2}$$

$$\sum_{j=1}^{n} X_{tij} = 1, \; i = 1\ldots.n, \; t = 1,\ldots., P \tag{3}$$

$$Y_{tijl} = X_{(t-1)ij} * X_{til}, \; i, j, l = 1,\ldots n; t = 2,\ldots, P \tag{4}$$

where

P: Number of periods in planning horizon

n: Number of machines in the layout

i,k: Machines in the layout

j,l: Locations in the layout

f_{ik}: Flow cost from machine i to k

d_{jl} : Distance from location j to l

Y_{tijl}: 0,1 variable for shifting i from j to l in period t

X_{tij}: 0,1 variable for locating machine i at location j in period t

A_{tijl}: Cost of shifting from j to l in period t. (A_{tijj}= 0)

Equation (1) is the sum of the material flow cost and layout rearrangement cost for the planning horizon. Equations (2) and (3) state that each machine must be located and each location must be occupied in every period. Equation (4) states that the 0-1 departmental rearrangement variable takes on a value of 1 only if the machine shifts its location at the end of a period.

There can be N! possible layout arrangements, for N machines. Enumeration of which is a very difficult approach to solving the QAP. However, as the problem size (N) increases, the number of possible arrangements (N!) also increases. This approach aims for good solution, which is indicated by solution obtained in the shortest processing time and comparable to results obtained by other algorithms. The program is run on I3 computer with 4 GB RAM and 2.4 GHz processor; so the computation time is not considered here, as it is very small. It ranges from 2 to 10 seconds. Hence, the goodness of any solution approach is adjudged against known example problems, for which the best solutions have been claimed.

3.2 Real coded genetic algorithm

Goldberg (1991) has discussed the approach for binary and real coding of GA. GA imitates the mechanism of the natural selection and evolution and aims to solve an optimisation problem with object function $f(x)$ where $x = [x1\ x2\ \ldots\ldots\ xn]$ is the N-dimensional vector of optimisation parameters. It has proved to be an effective and powerful global optimisation algorithm for any combinatorial optimisation problems, especially for discrete optimisation parameters, non-differentiable and discontinuous object function.

Binary GA is made up of *genes* and *chromosomes*. The conventional binary GA encodes the optimisation parameters into binary code string. A gene in GA is a binary bit. RCGA possesses many advantages than its binary coded counterpart when dealing with continuous search spaces with large dimensions and a great numerical precision is required. Each gene represents a variable of the problem in RCGA, and the size of the chromosome is kept the same as the length of the solution to the problem. RCGA can deal with large domains without sacrificing precision unlike the binary implementation (assuming a fixed length for the chromosomes). In addition, RCGA possesses the capacity for the local tuning of the solutions; it also allows integrating the domain knowledge to improve the performance of GA.

The binary GA does not operate directly on the optimisation parameters but on a discretised representation of them. Discretisation error will inevitably be introduced when encoding a real number. The encoding and decoding operations also make the algorithm more computationally expensive for problems with real optimisation parameters. Both theoretical proof and practical experiences show that RCGA usually works better than binary GA, especially for problems with real optimisation parameters (Herrera, Lozano and Verdegay, 1998).

The RCGA operates on a population of chromosomes (or individuals, creatures, etc.) simultaneously. It starts from an initial population, generated randomly within the search space. Once the initialisation is completed, the population enters the main RCGA loop and performs a global optimisation for searching the optimum solution of the problem.

GA has some limitations such as all offspring are accepted and their parent strings are abandoned at the end of every generation regardless of their fitness values. Because of this, there is a risk of replacing good parent strings with deteriorated child string. However, in the methodology of RCGA presented in this paper, the good parent strings are also restored with the good child strings and the care is taken that both good parent and child strings will go to the next generation. Another limitation of GA is that only good parents are given chance to produce the offspring, this is also taken care in the methodology of RCGA, as roulette wheel selection is employed and the chance is given to all parent strings. Hence, the answers obtained are near optimal.

3.3 Algorithm for RCGA

Step 1: The parameters like probability of Crossover (p), Probability of Mutation (q), population size and termination criteria are defined.

Step 2**:** A set of possible solutions called the "Initial Population" is initialised such that the values range uniformly throughout the search space.

Step 3: The Fitness Values are calculated for each of the individuals and the population is sorted in the decreasing order of the fitness.

Step 4: Using the "Roulette Wheel" method of 'Selection', few individuals are selected to undergo crossover.

Step 5: according to Crossover Probability the 'Crossover' between two individuals produces two 'Child Chromosomes'.

Step 6: then using the 'Probability of Mutation', a few individuals are mutated.

Step 7: At the end of performing crossover and mutation, we obtain the crossover and the mutated child individuals. These along with the parent population form super set for the next generation population.

Step 8: Steps 3–7 are repeated until the 'Termination' condition is reached.

Step 9: The population obtained at the end of the specified number of iterations is the outcome of this process.

4 Application of algorithm

In this work, three examples are considered. Example 1 is a standard benchmark function for dynamic layout problem (Rosenblatt, 1986). Example 2 is taken from Baykasoglu and Gindy (2001), and the example 3 is real life case study, considered from a press tool industry.

4.1 Example 1

Standard benchmark function for dynamic layout problem with six machines and five periods given by Rosenblatt (1986) is shown in Table 1. It shows the flow between the machines. The distance between the machines is assumed to be one unit and the layout is of two rows and three columns. To this problem, RCGA is iterated for 100 generations and the population size is 50. Crossover probability is 0.7 and mutation probability is 0.2.

Table 1 Standard benchmark function

Period 1						
From			*To*			
1	0	63	605	551	116	136
2	63	0	635	941	50	191
3	104	71	0	569	136	55
4	65	193	622	0	77	90
5	162	174	607	591	0	179
6	156	13	667	611	175	0
Period 2						
From			*To*			
1	0	175	804	904	56	176
2	63	0	743	936	45	177
3	168	85	0	918	138	134
4	51	94	962	0	173	39
5	97	104	730	634	0	144
6	95	115	983	597	24	0
Period 3						
From			*To*			
1	0	90	77	553	769	139
2	168	0	114	653	525	185
3	32	35	0	664	898	87
4	27	166	42	0	960	179
5	185	56	44	926	0	104
6	72	128	173	634	687	0
Period 4						
From			*To*			
1	0	112	15	199	665	649
2	153	0	116	173	912	671
3	10	28	0	182	855	542
4	29	69	15	0	552	751
5	198	71	42	24	0	758
6	62	109	170	90	973	0
Period 5						
From			*To*			
1	0	663	23	128	119	50
2	820	0	5	98	141	66
3	822	650	0	137	78	91
4	826	570	149	0	93	151
5	915	515	53	35	0	177
6	614	729	178	10	99	0
Shifting cost for departments						
	887	964	213	367	289	477

Source: Rosenblatt, 1986

Table 2 shows the analysis of aforementioned problem, when solved with presented RCGA and the results obtained for the problem provided by Rosenblatt using Dynamic programming, and the results obtained by Ballou. For the aforementioned problem, if static layout is applied then the cost obtained is \$ 78,374; and as if the machines are not changing their positions, its rearrangement cost is zero.

Table 2 Result and analysis of optimisation for dynamic layout problem (1986)

Approach	*Machine positions*	*Machines rearrangement cost*	*Material handling cost (\$)*	*Total cost (\$)*
Static	For all time periods 1 2 3 4 5 6	-	15,403	78,374
		-	18,617	
		-	15,156	
		-	14,659	
		-	14,539	
	Subtotal	-	78,374	
Dynamic (by Rosenblatt using dynamic programming)	First time period 1 3 5 6 4 2	-	12,822	75,384
	Second time period 1 4 2 5 3 6	2,310	14,853	
	Third time period 1 5 3 2 4 6	1,833	13,172	
	Fourth time period 1 6 4 2 5 3	1,346	13,032	
	Fifth time period 3 2 6 4 1 5	3,197	12,819	
	Subtotal	8,686	66,698	
Ballou	-	-	-	72,525
Presented approach (RCGA)	First time period 1 3 5 2 4 6	-	12,894	71,187
	Second time period 1 3 5 2 4 6	-	15,356	
	Third time period 1 5 3 2 4 6	502	13,172	
	Fourth time period 1 5 3 2 6 4	844	13,188	
	Fifth time period 6 5 3 2 1 4	1,364	13,867	
	Subtotal	2,710	68,477	

However, for all time periods, the layout is non-optimum, so the material handling cost is high. The presented RCGA approach is giving 5.56% better results as compared to the reported answer. In reported answer, the material handling cost is less as compared to present approach; but the machines rearrangement cost is more in the reported answer, which is $75,384. So the total cost obtained ($ 71,187) by presented approach is less than the reported answer.

The aforementioned comparison shows that the dynamic layout approach is definitely saving the cost as compared to the static layout approach. In addition, it shows that machines rearrangement cost is equally important as that of material handling cost. Total cost is comprising of machines rearrangement cost and material handling cost.

4.2 Example 2

Here, the data provided by Conway and Venkataraman, solved by Bayakasoglu and Gindy is considered. Bayakasoglu and Gindy (2001) have applied SA approach to solve this problem. There are nine departments and five periods considered. The optimum answer reported by author is $ 592,029 without shifting cost and $ 607,421 with shifting cost. For the same problem, Conway and Venkataraman have reported $ 593,856 as the best answer without shifting cost and $ 608,904 with the shifting cost. They have applied CONGA (Conway and Venkataraman GA) approach. For the aforementioned data, the same answer is obtained by presented RCGA, i.e., $ 592,029 without considering the shifting cost and $ 607,451 with considering the shifting cost. Table 3 shows the comparison of the solutions for example 2.

Table 3 Analysis of the solutions for example 2

	Without shifting cost ($)	*With shifting cost ($)*
Presented RCGA approach	5,92,029	6,07,421
SA approach	5,92,029	6,07,421
CONGA approach	5,93,856	6,08,904

Tables 2 and 3 indicate that the presented RCGA approach is working properly and giving better solutions. Hence, the presented RCGA approach is applied to the industry case study as given below.

4.3 Example 3: Industry case

Using the presented RCGA, a real case study from industry is solved. The company has total 18 different types of presses; details are shown in Table 4. Table 5 shows the product details. All the presses in the industry are mounted on vibration absorbing pads and have no permanent foundations. All the locations for the machines are equipped with necessary electrical fittings and compressor points. It is assumed that the product mix and quantity for future time periods are known. Figure 1 shows the initial layout of plant. The machine positions are shown in it and the distances between the machines are in the Foot. The matrix of rectilinear distances between machines is shown in Table 6. The operation sequence of the different jobs and their quantities at the respective time periods are shown in Tables 7–11. This case study is solved for five time periods. The shifting cost is same for all machines and is $12.5.

Table 4 Press capacities

Serial number	*Machine name*	*Assigned name*	*Capacity (Ton)*
1	PP02	P1	50
2	PP06	P2	100
3	PP03	P3	75
4	PP04	P4	50
5	PP10	P5	50
6	PP01	P6	30
7	PP14	P7	75
8	PP13	P8	150
9	PP05	P9	125
10	PP20	P10	100
11	PP17	P11	200
12	PP15	P12	150
13	PP08	P13	150
14	PP07	P14	150
15	PP11	P15	150
16	PP12	P16	150
17	PP09	P17	150
18	PP19	P18	200

Table 5 Details of products

Serial no.	*Product no.*	*Assigned name*	*Name of part*	*Operations*	*Press*
1	3236907	J1	M D I Carrier plate	Blanking	150 T
				I window piercing	150 T
				II window piercing	125 T
				Shaving	150 T
				16 hole piercing	150/100 T
				Planishing	150 T
2	RA0500081571	J2	Bottom plate	Blanking	100T
				Hole piercing	75T
				Forming	100T
3	RA0400081571	J3	Top plate	Blanking	150T
				Hole piercing	75 T
				Forming	100 T
4	3233924/020	J4	MDI Turbo rear cover	Blanking	150 T
				Six hole and six window piercing	150 T
				Embossing	200 T
				Forming	150 T
				Planishing	150 T

Table 5 Details of products (continued)

Serial no.	*Product no.*	*Assigned name*	*Name of part*	*Operations*	*Press*
5	3233925/020	J5	MDI Turbo front cover	Blanking	150T
				Six hole and six window piercing	150T
				Monogram stamping	50T
				Embossing and stamping	200 T
				Forming	150 T
				Planishing	150 T
6	0102AG0920/930N	J6	Renif panel front to sd Eng. Comp. LH/RH	First draw	150 T
				First trimming	150 T
				Second trimming	150 T
				Planishing	150 T
7	RA0405000041	J7	Bracket Dao-41 powder coated	Blanking	100 T
				Forming	125 T
				Slot piercing	75 T
8	RAO700100118	J8	Main plate LH/RH-support plate	Blanking	50 T
				Forming	50 T
9	459 10090	J9	Control arm	Blanking and hole piercing	150 T
				Rib embossing	150 T
				Shaving	150 T
				Bending	150 T
				Clinching	30/50 T
10	8459230	J10	PNL Wheel Arch Cargo Box-4 Door	Cropping	75 T
				I Bending	manually
				II Bending	manually
11	180N	J11	Relief fender gusset RH	Blanking and hole piercing	75 T
				Forming	100 T
12	3233902/020	J12	91/8 Reg. front cover	Blanking	100 T
				First draw	150 T
				First trimming	150 T
				Second trimming	150 T
13	3495903	J13	Bearing retainer msl	Blanking	50 T
				First forming	150 T
				Second forming	30 T
				Flaring	30 T
				Trimming	30 T

Table 5 Details of products (continued)

Serial no.	*Product no.*	*Assigned name*	*Name of part*	*Operations*	*Press*
14	3233901/020	J14	12"&14"N.D.Cover	Blanking	150 T
				8 Window and hole pea	150 T
				Forming	150 T
				Planishing	150 T

Figure 1 Initial layout of the company (use online version for colours)

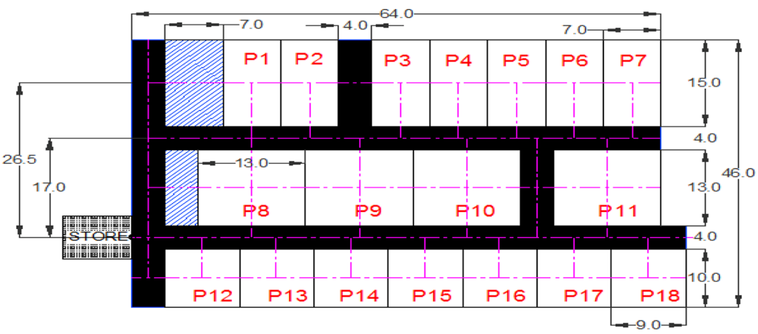


Table 6 Rectilinear distances between machines

M/C	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*
1	0	26	37	44	51	58	65	18	31	44	61	52.5	61.5	70.5	79.5	72.5	72.5	81.5
2		0	30	37	44	51	58	25	24	37	54	59.5	68.5	77.5	74.5	65.5	65.5	74.5
3			0	26	33	40	47	36	23	26	43	70.5	79.5	72.5	63.5	54.5	54.5	63.5
4				0	26	33	40	43	30	19	36	77.5	74.5	65.5	56.5	47.5	47.5	56.5
5					0	26	33	50	37	24	29	76.5	67.5	58.5	49.5	40.5	40.5	49.5
6						0	26	57	44	31	22	78.5	69.5	60.5	51.5	42.5	42.5	51.5
7							0	64	51	38	21	85.5	76.5	67.5	58.5	49.5	49.5	58.5
8								0	30	43	60	21.5	18.5	27.5	36.5	45.5	54.5	63.5
9									0	30	47	34.5	25.5	16.5	23.5	32.5	41.5	50.5
10										0	34	47.5	38.5	29.5	20.5	19.5	28.5	37.5
11											0	64.5	55.5	46.5	37.5	28.5	19.5	20.5
12												0	23	32	41	50	59	68
13													0	23	32	41	50	59
14														0	23	32	41	50
15															0	23	32	41
16																0	23	32
17																	0	23
18																		0

Table 7 Sequence of operations and job quantities at time period 1

Job no.	*Sequence of operations*					*Quantity per period*
J1	P14	P15	P16	P15	P14	212650
J2	P8	P1	P8			236000
J3	P13	P7	P5			236000
J4	P8	P16	P6	P15	P16	193300
J5	P8	P16	P6	P15	P16	191000
J6	P14	P14	P14	P14		93500
J7	P2	P8	P7			167000
J8	P15	P3	P13			41500
J9	P11	P16	P15	P8	P5	170000
J10	P4	P4				19050
J11	P4	P16	P8			118000

Table 8 Sequence of operations and job quantities at time period 2

Job no.	*Sequence of operations*					*Quantity per period*
J1	P14	P15	P16	P15	P14	90000
J2	P8	P1	P8			125000
J3	P13	P7	P5			170000
J4	P8	P16	P6	P15	P16	100000
J5	P8	P16	P6	P15	P16	135000
J6	P14	P14	P14	P14		180000
J7	P2	P8	P7			195000
J9	P11	P16	P15	P8	P5	175000
J11	P4	P16	P8			160000
J12	P7	P11	P14	P9		177000
J13	P9	P16	P8	P15	P17	120000
J14	P13	P6	P5	P14		75000

Table 9 Sequence of operations and job quantities at time period 3

Job no.	*Sequence of operations*					*Quantity per period*
J1	P14	P15	P16	P15	P14	156650
J2	P8	P1	P8			236000
J3	P13	P7	P5			123000
J4	P8	P16	P6	P15	P16	140300
J5	P8	P16	P6	P15	P16	191000
J6	P14	P14	P14	P14		163500
J7	P2	P8	P7			167000
J8	P15	P3	P13			41500

Table 9 Sequence of operations and job quantities at time period 3 (continued)

Job no.	*Sequence of operations*					*Quantity per period*
J9	P11	P16	P15	P8	P5	170000
J10	P4	P4				19050
J11	P4	P16	P8			118000
J12	P7	P11	P14	P9		160000

Table 10 Sequence of operations and job quantities at time period 4

Job no.	*Sequence of operations*					*Quantity per period*
J1	P14	P15	P16	P15	P14	90000
J2	P8	P1	P8			125000
J3	P13	P7	P5			170000
J4	P8	P16	P6	P15	P16	100000
J5	P8	P16	P6	P15	P16	135000
J6	P14	P14	P14	P14		180000
J7	P2	P8	P7			195000
J8	P15	P3	P13			41500
J9	P11	P16	P15	P8	P5	175000
J11	P4	P16	P8			160000
J12	P7	P11	P14	P9		177000
J13	P9	P16	P8	P15	P17	120000
J14	P13	P6	P5	P14		75000

Table 11 Sequence of operations and job quantities at time period 5

Job no.	*Sequence of operations*					*Quantity per period*
J1	P14	P15	P16	P15	P14	127700
J2	P8	P1	P8			115000
J3	P13	P7	P5			150300
J4	P8	P16	P6	P15	P16	110000
J5	P8	P16	P6	P15	P16	135000
J6	P14	P14	P14	P14		180000
J7	P2	P8	P7			195000
J9	P11	P16	P15	P8	P5	145000
J12	P7	P11	P14	P9		177000
J13	P9	P16	P8	P15	P17	120000
J14	P13	P6	P5	P14		115000

4.4 Working of RCGA

As discussed in Section 3, the working of RCGA in the presented approach is explored as follows:

Step 1: The parameters taken are *Population = 50*, *Generations = 100*, *Crossover Probability = 0.7, Mutation Probability 0.2*. Population size is determined by various experimentations. Different sizes from 10 to 100 are checked and it is observed that after population size 50, change in result is insignificant. In addition, generation number is determined by various experimentations and it is observed that after 100 generations no significant change is observed. The crossover probability and mutation probability are determined for various combinations, and it is showed in Table 17.

Step 2: As there are 18 machines, assume that each machine is known by a number and all machines are numbered from 1 to 18. Now a random number is generated, consisting of 1 to 18 numbers randomly and number of time period times (i.e., five times in this problem). This random number is the sequence of machines for the five time periods in the layout. This set of sequence is the one *Chromosome*. In this way, 50 *Chromosomes* are generated randomly. It is called as the *Initial Population*. Following is the example of the chromosome for aforementioned explanation.

10 9 18 15 8 6 13 4 1 16 3 2 12 17 5 11 14 7 / 1 7 9 14 15 5 17 18 8 16 11 10 3 2 13 4 12 6 / 2 6 17 10 8 16 1 12 3 15 4 13 9 11 14 5 18 7 / 15 16 17 2 14 11 7 6 1 8 5 4 3 10 12 9 18 13 / 10 9 11 14 6 8 7 13 5 15 16 18 4 3 12 2 1 17

Step 3: According to Eq. (1), the *Fitness Function (Z)* is calculated and is shown in Table 12. Here, the rectilinear distance between machines and the quantity to be moved between those machines and the cost per piece per feet are considered to calculate *Z*. Because of the space constraints, only six chromosomes are shown in Table 12.

Step 4: Using *Roulette Wheel* method few individuals are selected to undergo the *Crossover*, which is shown in Table 13. With the sketch shown below, a simple example is explained.

Table 12 Fitness function calculated according to Eq. (2)

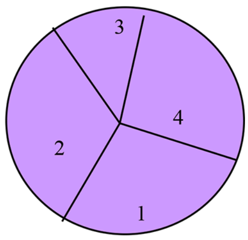
Serial no.	*Initially populated chromosome*	*Fitness function (Z)*
1	10 9 18 15 8 6 13 4 1 16 3 2 12 17 5 11 14 7 / 1 7 9 14 15 5 17 18 8 16 11 10 3 2 13 4 12 6 / 2 6 17 10 8 16 1 12 3 15 4 13 9 11 14 5 18 7 / 15 16 17 2 14 11 7 6 1 8 5 4 3 10 12 9 18 13 / 10 9 11 14 6 8 7 13 5 15 16 18 4 3 12 2 1 17	901772
2	6 16 1 4 3 7 13 15 8 9 18 14 12 10 17 11 5 2 / 12 3 9 5 11 1 4 18 17 7 2 10 6 14 13 8 16 15 / 16 6 14 18 3 5 13 15 11 9 7 4 2 17 12 10 8 1 / 10 7 13 6 15 8 3 1 11 16 9 4 2 12 5 14 17 18 / 2 9 7 11 15 16 14 12 18 8 10 3 17 1 4 6 5 13	914030

Table 12 Fitness function calculated according to Eq. (2) (continued)

Serial no.	*Initially populated chromosome*	*Fitness function (Z)*
3	11 14 5 15 8 13 18 17 12 16 7 3 4 10 2 1 6 9 / 3 17 9 12 8 13 5 6 16 15 7 18 4 10 2 14 1 11 / 16 15 11 9 5 13 12 6 14 8 1 3 10 17 4 2 18 7 / 13 12 15 6 8 16 7 3 5 14 11 18 1 10 17 2 9 4 / 13 18 11 15 5 17 6 8 7 16 14 12 2 1 10 3 9 4	896916
4	10 9 7 5 16 15 6 4 13 8 1 11 2 12 18 3 14 17 / 3 9 17 15 1 6 11 2 14 8 16 18 10 7 4 12 5 13 / 17 11 14 8 15 7 1 5 3 16 6 4 9 12 13 10 2 18 / 3 5 7 8 15 18 10 13 11 16 6 14 12 17 1 2 9 4 / 13 2 15 3 14 1 9 6 16 5 8 17 12 18 10 4 11 7	894025
5	12 6 5 16 8 3 10 1 14 15 11 4 17 13 2 7 9 18 / 3 1 4 15 6 17 10 13 16 8 2 11 18 14 5 12 7 9 / 14 4 3 18 10 12 7 9 5 11 13 6 15 16 1 8 2 17 / 2 7 11 4 8 14 9 5 1 16 3 12 13 6 17 15 18 10 / 1 4 9 2 11 15 6 18 14 8 16 17 3 7 5 12 10 13	910061
6	9 10 6 15 16 8 2 4 11 14 5 3 18 12 13 7 17 1 / 11 4 6 14 8 5 17 16 2 15 7 12 3 10 18 9 1 13 / 12 6 18 8 15 14 1 11 10 16 2 9 3 17 7 5 4 13 / 17 10 13 8 16 9 3 6 2 15 7 1 18 12 4 11 14 5 / 1 8 4 16 14 6 11 2 12 15 5 17 10 3 7 13 9 18	891021

Figure 2 Roulette wheel's example (Islier, 1998) (see online version for colours)

	Chromosome	Fitness
1	2 4 3 5 6 1	0.0752
2	3 5 6 1 4 2	0.1688
3	1 4 6 3 2 5	0.0311
4	2 4 6 1 5 3	0.0528



Suppose there are four *Chromosomes* generated and their *Fitness* values are found as shown above. The aforementioned wheel shows the contribution of each chromosome if the total of all is considered. When the wheel is spun, the probability of getting selected of 1, 2 and 4 *Chromosome* is more as compared to *Chromosome* 3. Therefore, the output may be as follows:

Figure 3 Output of Roulette wheel (Islier, 1998) (see online version for colours)

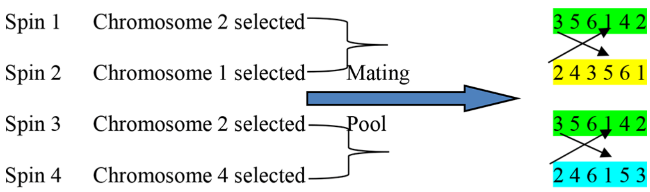


Table 13 Selection of individuals by Roulette wheel method

Serial no.	*Initially populated chromosome*	*Fitness function (Z)*
1	9 10 6 15 16 8 2 4 11 14 5 3 18 12 13 7 17 1 / 11 4 6 14 8 5 17 16 2 15 7 12 3 10 18 9 1 13 / 12 6 18 8 15 14 1 11 10 16 2 9 3 17 7 5 4 13 / 17 10 13 8 16 9 3 6 2 15 7 1 18 12 4 11 14 5 / 1 8 4 16 14 6 11 2 12 15 5 17 10 3 7 13 9 18	891021
2	9 10 6 15 16 8 2 4 11 14 5 3 18 12 13 7 17 1 / 11 4 6 14 8 5 17 16 2 15 7 12 3 10 18 9 1 13 / 12 6 18 8 15 14 1 11 10 16 2 9 3 17 7 5 4 13 / 17 10 13 8 16 9 3 6 2 15 7 1 18 12 4 11 14 5 / 1 8 4 16 14 6 11 2 12 15 5 17 10 3 7 13 9 18	891021
3	11 14 5 15 8 13 18 17 12 16 7 3 4 10 2 1 6 9 / 3 17 9 12 8 13 5 6 16 15 7 18 4 10 2 14 1 11 / 16 15 11 9 5 13 12 6 14 8 1 3 10 17 4 2 18 7 / 13 12 15 6 8 16 7 3 5 14 11 18 1 10 17 2 9 4 / 13 18 11 15 5 17 6 8 7 16 14 12 2 1 10 3 9 4	896916
4	10 9 7 5 16 15 6 4 13 8 1 11 2 12 18 3 14 17 / 3 9 17 15 1 6 11 2 14 8 16 18 10 7 4 12 5 13 / 17 11 14 8 15 7 1 5 3 16 6 4 9 12 13 10 2 18 / 3 5 7 8 15 18 10 13 11 16 6 14 12 17 1 2 9 4 / 13 2 15 3 14 1 9 6 16 5 8 17 12 18 10 4 11 7	894025
5	10 9 18 15 8 6 13 4 1 16 3 2 12 17 5 11 14 7 / 1 7 9 14 15 5 17 18 8 16 11 10 3 2 13 4 12 6 / 2 6 17 10 8 16 1 12 3 15 4 13 9 11 14 5 18 7 / 15 16 17 2 14 11 7 6 1 8 5 4 3 10 12 9 18 13 / 10 9 11 14 6 8 7 13 5 15 16 18 4 3 12 2 1 17	901772
6	12 6 5 16 8 3 10 1 14 15 11 4 17 13 2 7 9 18 / 3 1 4 15 6 17 10 13 16 8 2 11 18 14 5 12 7 9 / 14 4 3 18 10 12 7 9 5 11 13 6 15 16 1 8 2 17 / 2 7 11 4 8 14 9 5 1 16 3 12 13 6 17 15 18 10 / 1 4 9 2 11 15 6 18 14 8 16 17 3 7 5 12 10 13	910061

Step 5: Here the *Crossover* is done on 70% of chromosomes used above. Here the chromosome length is 18 × 5 = 90 numbers. In which 1–18 numbers are in random in first set and similarly in the next four sets. To perform the crossover, two parent strings are selected and a random number is generated in between 1 and 90. Suppose that number is 48. It falls in the third set of 1–18 numbers. From 49th number, the chromosomes are swapped to get the two offspring. However, the third set will have few numbers repeated which gives infeasible solution. Therefore, to overcome this repetition problem following procedure is adapted. Due to the space limit, following example demonstrates only the third set as said above. Selected chromosomes, i.e., *Parent 1* and *Parent 2* are shown in Figure 4.

Figure 4 Selected chromosomes (see online version for colours)

Parent 1	6	2	13	7	9	15	14	1	8	17	5	18	16	3	10	11	4	12
Parent 2	4	8	13	1	7	11	3	9	5	16	18	2	12	10	17	6	15	14

As the pivot point is 48, it will lie in third set at 12th position. So, the children obtained after the *Crossover* would be as per Figure 5.

Figure 5 Repetition of genes in chromosome after crossover (see online version for colours)

Child 1	4	8	13	1	7	11	3	9	5	16	18	2	16	3	10	11	4	12
Child 2	6	2	13	7	9	15	14	1	8	17	5	18	12	10	17	6	15	14

The *Child 1* and *Child 2* obtained after *Crossover* from *Parent 1* and *Parent 2* are shown above. As some of the *Genes* from children are repeated, it is shown by arrows. To overcome this difficulty, the following method is used. In the children's *Chromosome*, whichever *Genes* are repeated after *Pivot Point* are omitted and the non-repeating *Genes* are kept as it is, it is shown in Figure 6.

Figure 6 Chromosomes after removal of repeated genes (see online version for colours)

Child 1	4	8	13	1	7	11	3	9	5	16	18	2	-	-	10	-	-	12
Child 2	6	2	13	7	9	15	14	1	8	17	5	18	12	10	-	-	-	-

Then at the vacant places of that *Chromosome*, whichever the numbers are absent from 1 to 18 are filled in the sequence. So the final *Chromosome* for the *Child 1* and *Child 2* has become as shown in Figure 7.

Figure 7 Chromosomes after filling the genes at vacant positions (see online version for colours)

Child 1	4	8	13	1	7	11	3	9	5	16	18	2	6	14	10	15	17	12
Child 2	6	2	13	7	9	15	14	1	8	17	5	18	12	10	3	4	11	16

Step 6: Using the *Mutation Probability* 0.2, i.e., 20 % of the above crossed over *Chromosomes* are *Mutated*. Here it is demonstrated for only one set of chromosome. Here the two numbers are randomly generated in between 1 and 18, suppose they are 4 and 9. Then the *Genes* at that place from the *Chromosomes* are exchanged from their places.

i.e. 7 1 5 **17** 8 4 11 18 **2** 10 14 12 13 3 15 16 9 6

After mutation it will become

7 1 5 **2** 8 4 11 18 **17** 10 14 12 13 3 15 16 9 6

Step 7: After *Crossover* and *Mutation* as above, all crossed over and mutated *Chromosomes* along with some parent *Chromosomes* forms the new set of *Chromosome* for the next *Generation*.

Step 8: In this way, the numbers of *Generations* are taking place and at every *Generation* the best *Chromosome* values are stored.

After 100 trials, the best machine layout obtained so far by the present RCGA approach is shown in Table 14. Here, the static condition and dynamic conditions are compared.

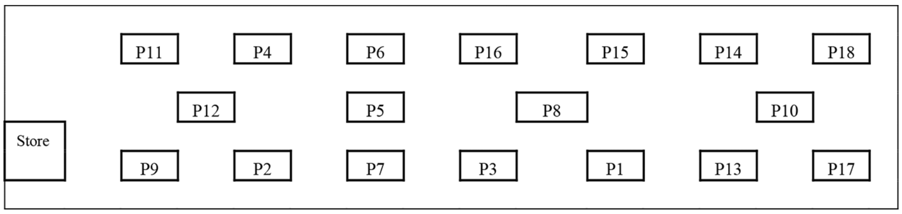
Table 14 Optimum machine layout obtained by presented RCGA

	First period		*Second period*		*Third period*		*Fourth period*		*Fifth period*		*Total cost ($)*
Static condition	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18										18,529
Dynamic condition	11	4	10	13	5	13	9	18	13	2	15,854*
	6	16	4	9	7	16	2	8	7	5	
	15	14	11	3	3	11	10	11	4	1	
	18	12	14	1	9	18	1	14	10	11	
	5	8	16	7	8	15	15	16	9	8	
	10	9	18	6	14	10	7	12	3	12	
	2	7	12	15	2	1	17	6	14	6	
	3	1	8	5	17	4	5	4	15	16	
	13	17	17	2	6	12	13	3	17	18	

*Total cost = Material handling cost + rearrangement cost, i.e., 13,625 = $ 14,779 + $ 1,075

Figure 8 shows the graphical representation of the proposed layout at time period 1, which clearly reveals the change in machine positions and so the dynamics of the layout.

Figure 8 The proposed layout at first time period



Analysis of aforementioned problem is shown in Table 15. If the original layout is kept as it is for all the five time periods then the total cost required is larger than the dynamic machine layout. Table 15 also indicates that the rearrangement cost required for static layout is zero; however, the material handling cost is more than the dynamic layout condition. In dynamic layout, the rearrangement cost is needed, but because of changed positions of the machines, the best layout at that respective time period is obtained and so the material handling cost is also lower. Ultimately, it results in the minimum total cost, which is the sum of rearrangement costs and material handling costs. The aforementioned analysis shows that the dynamic machine layout is saving 15.5% of the cost than the static machine layout.

It is observed that the dynamic layout is more suitable for the aforementioned industry sector in the changing environment, than the static layout. Even though the additional rearrangement cost is required in case of dynamic layout, the best layout combination obtained from first to fifth period's, results in low material handling cost, hence the total cost is less. The presented RCGA approach is run individually for 100 trials and its performance is shown in Table 16. The convergence of the presented algorithm is shown in Figure 9.

Table 15 Analysis of results for industry case

	Rearrange-ment cost ($) (a)	*First time period ($) (b)*	*Rearrange-ment cost ($) (c)*	*Second time period ($) (d)*	*Rearrange-ment cost ($) (e)*	*Third time period ($) (f)*	*Rearrange-ment cost($) (g)*	*Fourth time period($) (h)*	*Rearrangement cost ($) (i)*	*Fifth time period ($) (j)*	*Total ($) (a + b + c + d + e + f + g + h + i + j)*
Static layout	0	3,815	0	3,723	0	3,532	0	3,910	0	3,549	18,529
Dynamic layout	225	2,828	225	3,087	225	2,813	200	3,086	200	2,964	15,854

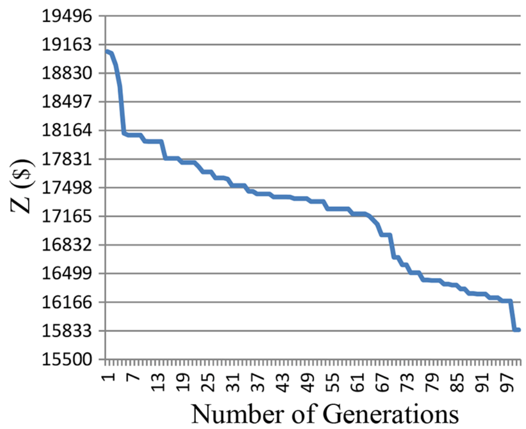
Table 16 Performance of presented RCGA

Runs	*Min total cost ($)*	*Max total cost ($)*	*Average ($)*	*Standard deviation (%)*
100	15,854	19,080	16,960	4.75

Table 17 Selection of combination of crossover and mutation probability

Set	*Crossover probability*	*Mutation probability*	*Standard deviation (%)*
Set - 1	0.6	0.1	3.58
Set - 2	0.6	0.15	4.13
Set - 3	0.6	0.2	3.2
Set - 4	0.7	0.1	3.73
Set - 5	0.7	0.15	3.1
Set - 6	**0.7**	**0.2**	**2.89**
Set - 7	0.8	0.1	5.05
Set - 8	0.8	0.15	3.98
Set - 9	0.8	0.2	4.21

Figure 9 Convergence graph for RCGA (see online version for colours)



5 Conclusions

Dynamic machine layout concept is useful in the industries where product mix and product quantity are changing with respect to time. In this work, dynamic machine layout approach is attempted for real life application in press working industries. The RCGA is applied. It is observed that the dynamic layout when optimised with RCGA provides a significant reduction of 15.5% in total cost over the existing layout. It clearly indicates the potential of this approach to solve dynamic layout problems for other industries as well. Moreover, it is also observed that the presented algorithm also provides high convergence rate and it converges to best obtained solution in 100 generations. On running the algorithm individually for 100 times, its minimum, maximum, and mean values are considered to find out the standard deviation, which is 4.75%. It shows the robustness of the algorithm for solving the dynamic layout problem.

Acknowledgement

The authors are grateful to Mr. Mukund Kulkarni, Director, S M Auto Stampings Pvt. Ltd., Nasik, (M.S.) India, for their support and cooperation to get real industrial data.

References

Ahin R. and Turkbey, O. (2008) 'A new hybrid tabu-simulated annealing heuristic for the dynamic facility layout problem', *International Journal of Production Research*, Vol. 47, pp.1–19, iFirst.

Ashtiani, B., Aryanezhad, M.B. and Moghaddam, B.F. (2007) 'Multi start simulated annealing for dynamic plant layout problem', *Journal of Industrial Engineering International*, Vol. 3, No. 4, pp.44–50.

Azimi, P. and Charmchi, H.R. (2012) 'A new heuristic algorithm for dynamic facility layout problem with budget constraint', *World Academy of Science, Engineering and Technology*, Vol. 65, pp.478–484.

Balakrishnan, J., Cheng, C.H., Conway, D.G. and Lau, C.M. (2003) 'A hybrid genetic algorithm for the dynamic plant layout problem', *International Journal of Production Economics*, Vol. 86, No. 2, pp.107–120.

Baykasoglu, A. and Gindy, N.N.Z. (2001) 'A simulated annealing algorithm for dynamic layout problem', *Computers and Operations Research*, Vol. 28, pp.1403–1426.

Bock, S. and Hoberg, K. (2007) 'Detailed layout planning for irregularly-shaped machines with transportation path design', *European Journal of Operational Research*, Vol. 177, pp.693–718.

Brunese, P.A. and Tanchoco, J.M.A. (2012) 'On implied within-building constraints for machine layout', *International Journal of Production Research*, Vol. 51, No. 6, pp.1937–1952.

Donga, M., Wu, C., Hou, F. (2009) 'Shortest path based simulated annealing algorithm for dynamic facility layout problem under dynamic business environment', *Expert Systems with Applications*, Vol. 36, pp.11221–11232.

Goldberg, D.E. (1991) 'Real Coded Genetic Algorithms, Virtual Alphabets and Blocking', *Complex Systems,* Vol. 5, pp.139–167.

Goncalves, E.V. and Tiberti, A.J. (2006) 'A group genetic algorithm for the machine cell formation problem', *International Journal of Production Economics*, Vol. 102, pp.1–21.

Herrera, F., Lozano, M., Verdegay, J. L. (1998) 'Tackling Real coded Genetic algorithms: Operators and tools for behavioral analyses, *Artificial Intelligence Review*, Vol. 12, pp.265–319.

Islier, A.A. (1998) 'A genetic algorithm approach for multiple criteria facility layout design', *International Journal of Production Research*, Vol. 36, no. 6, pp.1549–1569.

Jinde, A.R. and Telsang, M.T. (2014) 'Material handling cost reduction by redesigning layout - a genetic approach', NCIETM held at NITIE Mumbai, pp.174–181.

Krishnan, K. K., Cheraghi, S.H., Nayak, C. (2008) 'Case study in using in dynamic from between charts to solve dynamic facility layout problems', *California Journal of Operations Management*, Vol. 6, No. 1, pp.115–122.

Lenin, N., SivaKumar, M., Islam, M. N., Ravindran, D. (2012) 'Multi-objective optimization in single-row layout design using a genetic algorithm', *International Journal of Advanced Manufacturing Technology*, Vol. 170, no. 12, pp.4608.

Maiti A. K., Bhunia A. K., M. Maiti (2006) 'An application of real-coded genetic algorithm (RCGA) for mixed integer non-linear programming in two-storage multi-item inventory model with discount policy.' *Applied Mathematics and Computation.* Vol. 183, Issue 2, Pages 903–915. Elsevier.

McKendall, A.R. and Hakobyan, A. (2010) 'Heuristics for the dynamic facility layout problem with unequal-area departments', *European Journal of Operational Research*, Vol. 201, pp.171–182.

Mohammadi, E. and Moghadam, M.A. (2011) 'Design of dynamic body shop layout in automobile industry', *International Journal of Automotive Engineering*, Vol. 1, Number 4.

Moslemipour, G. and Lee, T. S. (2012) 'Intelligent design of a dynamic machine layout in uncertain environment of flexible manufacturing systems', *Journal of Intelligent Manufacturing*, Vol. 23, pp.1849–1860.

Norouzi, A., Zolfaghari, A., Minuchehr, A. H., Khoshahval F. (2011) 'An enhanced integer coded genetic algorithm to optimize PWRs', Progress in Nuclear Energy 53, 449–456.

Pal P., Das C. B., Panda A. and Bhunia A. K. (2005). 'An application of real-coded genetic algorithm (for mixed integer non-linear programming in an optimal two-warehouse inventory policy for deteriorating items with a linear trend in demand and a fixed planning horizon).' *International Journal of Computer Mathematics,* Volume 82, Issue 2. (Taylor and Francis online).

Ranjan, R. and Pawar, P. J. (2014) 'Assembly Line Balancing Using Real Coded Genetic Algorithm', *International Journal of Scientific Research in Computer Science and Engineering*, Vol. 2, no. 4, pp.1–5.

Rosenblatt, M. J. (1986) 'The dynamics of plant layout', *Management science*, Vol. 32, no. 1.

Rosenblatt, M. J. and Golany, B. (1992) 'A distance assignment approach to the facility layout problem', *European Journal of Operational Research*, Vol. 57, pp.253–370.

Sirinaovakul, B. and Limudomsuk, T. (2007) 'Maximum weight matching and genetic algorithm for fixed-shape facility layout problem', *International Journal of Production Research*, Vol. 45, no. 12,pp.2655–2672.

Solimanpur, M. and Kamran, M. A. (2010) 'Solving facilities location problem in the presence of alternative processing routes using a genetic algorithm', *Computers & Industrial Engineering*, Vol. 59, pp.830–839.

Wang Yu-Fen, and Guo Xiau-Juan (2009). 'Real-coded Genetic Algorithm and Application in the Automatic Composing the Test Paper.' *Intelligent Information Technology Application*. IIT A 2009. IEEE X-plore digital library.