International Journal of Scientific Research in Computer Science and Engineering
Research Paper Volume-2, Issue-4 ISSN: 2320-7639

Assembly Line Balancing Using Real Coded Genetic Algorithm

Rajeev Ranjan[1*] and P.J. Pawar[2]

[1*,2]*K.K. Wagh Institute of Engineering Education and Research, Nasik, India*
ranjan_1614@yahoo.co.in; pjpawar1@rediffmail.com

Available online at www.isroset.org

Received: 10 Mar 2014 Revised: 22 Mar 2014 Accepted: 08 May 2014 Published: 31 Aug 2014

Abstract— Assembly line balancing is to assign the tasks to the workstations, so as to achieve the number of workstations and maximization of the production rate through reduction in balance delay. This paper represents the use of real coded genetic algorithm for assembly line balancing. An application example is presented and solved to illustrate the effectiveness of the presented algorithm. For the considered problem, tact time is fixed whereas the sequence of the work content can vary as per the precedence.

Keywords-*component ; Assembly line balancing, Real Coded genetic algorithm, Balance delay, Precedence fouling.*

I. INTRODUCTION

An assembly line is a manufacturing process usually used in the production of large amount of standardized products such as automobiles and household appliances. An assembly line consists of a sequence of stations connected with a material handling system such as conveyor system. At each station, certain tasks necessary to assemble a product are performed. Each station must complete the tasks within a fixed time period called the cycle time. An important decision problem in the assembly line is to determine the assignment of the tasks in order to optimize an objective function. This problem is called as assembly line balancing problem. Scholl and Becker [1] indicated that''the simple assembly line balancing problem (SALBP) concerns allocation of tasks among stations,so that precedence relations are not violated and a given objective function is optimized''.

The ALB problem falls into the NP-hard class of combinatorial optimization problems. If there are n tasks and r preference constraints, then there are n!/2r possible task sequences. Therefore, it can be time consuming for optimum seeking methods to gain an optimal solution within this extremely large search space for some manufacturing operations, such as car assembly lines with more than 100 workstations.

Despite the vast search space, many studies have tried to solve the ALB problem using optimum-seeking methods, such as linear programming, integer programming, dynamic programming and branch- and- bound approaches. However, none of these methods has proven to be of practical use for large assembly lines due to their computational inefficiency. Hence, the next research efforts have been directed towards the development of heuristic and meta-heuristics such as simulated annealing, Tabu search and genetic algorithms

Due to the complexity of the ALB problem, a growing number of researchers have employed genetic algorithms (GAs), and most industrial engineers also use them to optimize problems which are difficult to find an optimal solution for in a reasonable time. GAs provides an alternative to traditional optimization techniques by using directed random searches to locate optimum solutions in complex search spaces. Hence, because of the popularity of GAs' application to the ALB problem, some papers exist which review the subject, including Scholl and Becker [1] and Tasan and Tunali[2] have all tried to modify GA using modified selection techniques, individual representation, crossover techniques etc. in order to improve the algorithms.

In this work a methodology is proposed to solve assembly line balancing problem using modified real coded genetic algorithm. An application example is presented to show the effectiveness of the proposed approach.

II. LITERATURE REVIEW

It is revealed from the literature review that various algorithms/methods have been proposed by the researcher to solve the assembly line balancing problem. Amen [3] presented work on an exact method for cost-oriented assembly line balancing. A work on new heuristic method for mixed model assembly line balancing problem was published by Jina and Wub [4]. Fleszar and Hindi [5] presented a work on enumerative heuristic and reduction methods for the assembly line balancing problem. They presented a new heuristic algorithm and new reduction techniques for the type 1 assembly line balancing problem. A work on assembly line balancing in a mixed model sequencing environment with synchronous transfers was presented by Karabati and Sayin [6]. A fuzzy logic approach to assembly line balancing work was presented by Fonseca et al. [7]. Gokcen [8] presented a work on shortest route formulation of simple U-type assembly line balancing problem. A work was presented by Bukchin and Rabinowitch [9] on branch and bound based solution approach for the mixed-model assembly line balancing problem for minimizing stations and task duplication costs. Lapierre et al. [10] presented his work on balancing assembly lines with tabu search

Corresponding Author: *Rajeev Ranjan*

But the main issues with using them are, the results are trapped in local solution while the genetic algorithm has the ability to give optimal solution. As most heuristics are generally problem specific, their applications are limited. The focus of research thus shifts towards the development of powerful metaheuristic algorithms. A metaheuristic provides a general algorithmic framework which can be applied to various optimization problems. Classification of metaheuristic includes simulated annealing, tabu search, iterated local search and evolutionary computing. Evolutionary computing in general refers to several heuristic techniques based on the principles of natural evolution. One of these heuristics is genetic algorithms.

Our interest is in the application of genetic algorithms to combinatorial optimization problems. It is appropriate to start with an outline description of this type of approach to combinatorial optimization. Genetic algorithm has been use previously for many problem types. Genetic algorithm was used by Kuan and Mohamed [11] for solving Assembly Line Balancing problem with heuristics-treated initial population and produced required results. Darrel whitely [12] proved that the genetic algorithm can be used for various optimization fields. Edward and Micheal [13] used genetic algorithm for line balancing but they didn't consider the precedence fouling. The main motive to use genetic algorithm was to increase the amount of work content on each station and decreasing the number of work stations. Runwei and Tatsumi [14] used genetic for designing loop layout manufacturing systems. Alper & Gokalp [15] used genetic algorithm based approach for simultaneously balancing and sequencing of mixed-model U-lines with parallel workstations and zoning constraints. A work was presented by Simaria and Vilarinho [16] on genetic algorithm based approach to the mixed-model assembly line balancing problem of type II. Levitin et al. [17] works on genetic algorithm for robotic assembly line balancing

The present study is based on the using real coded genetic algorithm for line balancing by reducing the balance delay and considering the precedence foul. The problem considered here is having tacts with fixed time. The objective is to reduce the number of tacts utilized for production and thus saving manpower if possible. Next sections present the use of genetic algorithm for line balancing.

III. OPTIMIZATION MODEL

For line balancing there are some important factors required to control. Balance delay and precedence fouling are the most important criteria to solve a line balancing problem in the real world. In this section we formulate a new mathematical model based on the balance delay and precedence foul. The proposed model deals with minimization of balance delay and precedence fouls to achieve the efficient line balancing. For precedence fouls to be more effective on our result, penalty of 10 is considered while calculation of our function value.

Minimize Z= (B+P×F)

Where, B= Balance delay

$$B = \frac{nC_a - T_{wc}}{nC_a} \quad (1)$$

C_a= Maximum loading of the tact

n= Number of tacts

T_{wc}= Total work content

F= Cumulative precedence fouling

P= Penalty for each fouling (i.e. 10)

IV. REAL CODED GENETIC ALGORITHM

When bit-string representations of integers are used, Gray coding is often employed. In this way, small changes in the integer can be readily affected through mutations or crossovers. This has been found to help prevent premature convergence at so called Hamming walls, in which too many simultaneous mutations (or crossover events) must occur in order to change the chromosome to a better solution.

Other approaches involve using arrays of real-valued numbers instead of bit strings to represent chromosomes. Results from the theory of schemata suggest that in general the smaller the alphabet, the better the performance, but it was initially surprising to researchers that good results were obtained from using real-valued chromosomes. This was explained as the set of real values in a finite population of chromosomes as forming a virtual alphabet (when selection and recombination are dominant) with a much lower cardinality than would be expected from a floating point representation

The strengths of real-coded genetic algorithms include:

A. Increased efficiency: bit strings do not need to be converted to real numbers for every function evaluation.
B. Increased precision: since a real-number representation is used, there is no loss of precision due to the binary representation.
C. Greater freedom to use different mutation and crossover techniques based on the real representation.

V. APPLICATION EXAMPLE

Now an application example is considered to demonstrate and validate the presented real coded genetic algorithm for the balancing of assembly line. For balancing of work content only precedence constraint and tact time of tacts is considered. It is assumed that all the work content can be done on any tact and doesn't have any tact bound limitations. Details of the mentioned assembly line is as follows

A. Tact time considered = 20 units

B. Work elements with elemental timings are as shown in Table I.

TABLE I. ELEMENTAL TIME

Element	Time	Precedence
1	5	
2	3	1
3	7	1
4	5	1
5	6	3
6	3	3,4
7	4	2
8	6	5,6
9	8	4
10	9	7,8,9

C. Precedence Diagram

We have ten tasks with own times shown in table, the precedence graph (matrix) are presented and the cycle time is equal to 56 units. Every element has got definite elemental time

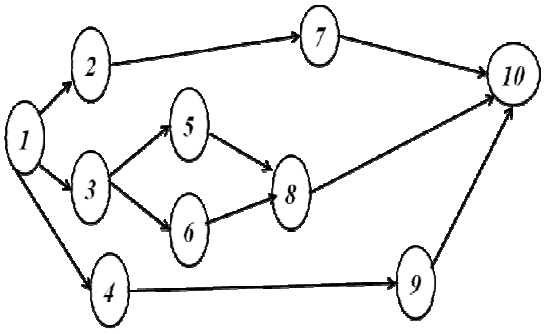

FIGURE I . PRECEDENCE DIAGRAM

VI. RGCA APPLICATION TO THE PROBLEM

Sabuncuoglu et al. [18] has introduced task-based representation (TBR) where each task is represented by a number that is placed on a string (i.e. individual) with the string size equal to the number of tasks. The tasks are ordered by the individual relative to their order of processing. The tasks are allocated to workstations so that the sum of the task times in each station does not exceed the cycle time.

A. Initial Population Selection

The initial population is generated randomly assuring feasibility according to precedence relations. So, all individuals in the population in all generational steps will be feasible.

TABLE II. INITIAL POPULATION

1	10	2	3	9	5	6	7	8	4	1
2	10	2	3	7	5	6	4	8	9	1
3	6	2	10	4	5	1	7	8	9	3
4	1	10	3	4	5	6	7	8	9	2
5	7	2	3	9	5	6	1	8	4	10
6	1	2	3	9	5	6	7	8	4	10
7	8	2	3	7	5	6	4	1	9	10
8	1	2	3	4	5	6	7	8	9	10
9	9	2	10	4	5	6	7	8	1	3
10	1	2	3	9	5	6	7	8	4	10

B. Crossover

The crossover is a process by which a string is divided into segments, which are exchanged with the segments corresponding to another string. With these process two new strings different to those that produced they are generated. It is necessary to clarify that the choice of strings crossed inside those that were chosen previously in the reproduction process is random .From the point of view of problem optimization; it is equal to the exploitation of an area of the parameters space.

The following outline shows the crossover process. Obtained sequences after crossover operation are shown in table

Before crossover (↓ ↓) → After crossover

3 10 8 4 1 2 9 5 6 7 → 3 10 8 10 8 1 7 5 4 6
3 9 2 10 8 1 7 5 4 6 → 3 9 2 4 1 2 9 5 6 7

Replacing with the missing numbers in the series
3 2 9 10 8 1 7 5 4 6
3 8 10 4 1 2 9 5 6 7
Now the series contains all the elements. And now mutation can be carried on the population

C. Mutation Technique

As with biological systems the mutation is manifested with a small change in the genetic string of the individuals. In the case of artificial genetic strings, the mutation is equal to a change in the elementary portion (allele) of the individuals' code. The mutation takes place with characteristics different to those that the individuals had at the beginning, characteristics that didn't possibly exist in the population. From the point of view of problem optimization, it is equal to a change of the search area in the parameters space. The above mentioned is illustrated with the following outline. Obtained sequences after mutation operation are shown in table.

Before mutation
3 2 9 10 8 1 7 5 4 6
3 8 10 4 1 2 9 5 6 7

After mutation
3 2 9 6 8 1 7 5 4 10
→ 3 10 8 4 1 2 9 5 6 7

D. Stopping Criteria and genetic algorithm settings

As the balance delay given by the conventional line balancing techniques used is 0.07, hence to obtain a better result the stopping criteria should be better than that.

- Population size: 10
- Selection rule: roulette wheel selection
- Crossover rate: 0.8
- Crossover technique: Two point crossover

E. Selection

Random numbers are generated and with respect to that relative population with fitness value are selected. Related values are copied for next iteration and fitness function is calculated on its basis. The roulette wheel selection method is used for the selection.

VII. COMPUTATIONAL RESULTS AND COMPARISON

The proposed GA is coded in Office Excel and executed on a Windows-based, Intel I5, 2.3GHz computer. Visual Basic on excel is chosen as the development platform because the availability of the toolbox and useful built-in functions. The proposed GA is tested with proposed methodology for the test problem to compare its output result with results obtained from conventional technique. The performances of problem are assessed by 2 criteria: (i) the number of workstations and (ii) realized loading of work content on each tact. Once the output result falls in our desired range we can plot the utilization diagram of the operator and compare the increase utilization. This section compares all the aspects of the results obtained.

TABLE III. WORK DISTRIBUTION

1	3	2	4	5	6	7	8	9	10	20	0.067
5	7	3	5	6	3	4	6	8	9		
5	12	15	20	6	9	13	19	8	17	20	
7	3	5	-14	3	4	6	-11	9	-17		
0	0	0	1	0	0	0	1	0	1	3	
0	0	0	0	0	0	0	1	0	3	0.067	
1	2	2	2	3	3	4	5	6	7		
0	0	0	0	0	0	0	0	0	0	0	

1st Row: It indicates the sequence elements after crossover and mutation
2nd Row: It indicates the relative elemental time of the elements in the 1st row
3rd Row: It indicates how many elements can be clubbed in a tact accumulating the tact time
5th Row: It indicates the total number of tacts to be used for the given elements
6th Row: It indicates the difference between the tact time and total cycle time at particular tact
7th Row: It indicates the relative precedence relation of the elements
8th Row: It indicates the precedence fouling of the elements in the assembly sequence

A. Calculation of functional value

Function value, Z = (B+P×F)
Where,

$$B = \frac{nC_a - T_{wc}}{nC_a}$$

Maximum loading of the tact, $C_a = 20$

Number of tacts, n= 3

Total work content, $T_{wc} = 56$

Cumulative precedence fouling, F= 0

Penalty for each fouling (i.e. 10), P=10

$$= \frac{3*(20) - 56}{3*20} = 0.067$$

Z= 0.067+10*0 = 0.067

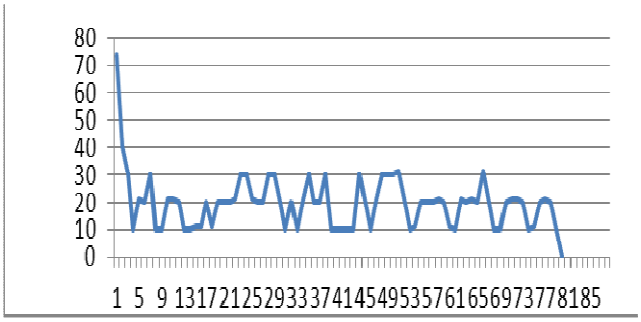

Figure II. FUNCTIONAL VALUE(Y-axis) Vs ITERATIONS (X-axis)

Comaparison of the results for different criterias are :

TABLE IV. SUMMARY SHEET

Criteria	Results of optimization using GA
No of operators	3
No of tacts used	3
Balance Delay	0.067
Maximum Loading of tact	20

VIII. CONCLUSION

In this paper, an assembly line balancing problem is solved using real coded genetic algorithm (RCGA) considering the constraint on precedence of operations. It is observed through the sample case of study that proposed approach can handle the problem of assembly balancing effectively. It can also be seen that the solution of the problem converges to the optimal solution in very few iterations (about 80-100).

REFERENCES

[1] C. Becker and A. Scholl, "A survey on problems and methods in generalized assembly line balancing," European Journal of Operational Research, vol. 168, pp. 694-715, Feb 1 2006

[2] S. O. Tasan & S. Tunali, "A review of the current applications of genetic algorithms in assembly line balancing," Journal of Intelligent Manufacturing, vol. 19, pp. 49-69, Feb 2008

[3] Matthias Amen,"An exact method for cost oriented assembly line balancing". International Journal of Production Economics, Vol. 64, pp.187-195,2000

[4] Mingzhou Jina & S. David Wub, "A new heuristic method for mixed model assembly line balancing problem". Computers & Industrial Engineering, Vol. 44, pp.159–169,2002

[5] Krzysztof Fleszar & Khalil S. Hindi , "An enumerative heuristic and reduction methods for the assembly line balancing problem". European Journal of Operational Research ,Vol. 145, pp. 606–620,2003

[6] Selcuk Karabati & Serpil Sayin, "Assembly line balancing in a mixed-model sequencing environment with synchronous transfers". European Journal of Operational Research, Vol. 149, pp.417–429, 2003

[7] D.J. Fonseca, C.L. Guest, M. Elam & C.L. Karr "A Fuzzy Logic Approach to Assembly Line Balancing". Mathware & Soft Computing, Vol.12, pp. 57-74,2005

[8] Hadi Gokcen, Kursat Agpak, Cevriye Gencer & Emel Kizilkaya , "A shortest route formulation of simple U-type assembly line balancing problem". Applied Mathematical Modelling ,Vol.29, pp.373–380,2005

[1] Yossi Bukchin & Ithai Rabinowitch,"Production, manufacturing and Logistics A branch-and-bound based solution approach for the mixed-model assembly line-balancing problem for minimizing stations and task duplication costs". European Journal of Operational Research ,Vol.174, pp. 492–508,2006

[2] Sophie D. Lapierre, Angel Ruiz & Patrick Soriano, "Balancing assembly lines with tabu search". European Journal of Operational Research ,Vol.168,pp.826–837,2006

[3] Kuan Eng Chong, Mohamed K. Omar & Nooh Abu Bakar, "Solving Assembly Line Balancing Problem using Genetic Algorithm with Heuristics-Treated Initial Population". World Congress on Engineering,Vol II,pp. 2-5,2008

[4] Darrell Whitley, " A genetic algorithm tutorial", Statistics and computing,Vol.4,pp.65-85,1994

[5] Edward J. Anderson & Michael C. Ferris, "A Genetic algorithm for the assembly line problem" , Computer Science Technical Report, 926,pp. 3-16,1990

[6] Runwei Cheng, Mitsuo Gen t, Tatsumi Tosawa, "Genetic algorithm for designing loop layout manufacturing systems", Computer Industrial Engineering, Vol 31,pp.587-591,1996

[7] Alper Hamzadayi & Gokalp Yildiz, "A genetic algorithm based approach for simultaneously balancing and sequencing of mixed-model U-lines with parallel workstations and zoning constraints". Computers & Industrial Engineering, Vol. 62, pp. 206-215,2012

[8] Ana Sofia Simaria & Pedro M. Vilarinho (2004) "A genetic algorithm based approach to the mixedmodel assembly line balancing problem of type II". Computers & Industrial Engineering, Vol. 47, pp.391–407,2004

[9] Gregory Levitin, Jacob Rubinovitz & Boris Shnits, "A genetic algorithm for robotic assembly line balancing". European Journal of Operational Research ,Vol.168, pp.811–825,2006

[10] I. Sabuncuoglu, et al., "Assembly line balancing using genetic algorithms," Journal of Intelligent Manufacturing, vol. 11, pp. 295-310, May 2000

