GENETIC ALGORITHM WITH TOPOLOGY TO SOLVE MINIMIZING WAITING TIME OF ALBP

Dr. (Mrs) G.Nirmala,* and D.Ramprasad**
*Head and Associate Professor in Mathematics, PG and Research Department of Mathematics,
K.N.G. Arts College for Women (Autonomous), Thanjavur. Tamilnadu • Email-nirmalamanokar11@yahoo.com
**Assistant Professor in Mathematics, PG and Research Department of Mathematics,
A.V.V.M. Sri Pushpam College (Autonomous), Poondi, Thanjavur, Tamilnadu • Email-drprasad1981@gmail.com

ABSTRACT:
Assembly line balancing problem (ALBP) is a well-known combinatorial optimization problem in production and operations management area. Due to the NP-hard nature of the ALBP, many attempts have been made to solve the problem efficiently. In this study, Genetic algorithm (GA) is adopted to solve the ALBP with the objective of minimizing the waiting time in the workstation. The key issue in solving ALBP is how to generate a feasible task sequence which does not violate the precedence constraints. This task sequencing is a vital work to be solved prior assigning tasks to workstation. In order to generate only feasible solution, a repairing strategy is integrated in the GA procedure. The computational result shows that the proposed approach is capable to obtain feasible solution with minimum waiting time for a simple model assembly line.

Keywords : ***Genetic Algorithm, Assembly line balancing, task sequencing, waiting time.***

INTRODUCTION

An assembly line comprises a series of successive workstation connected together by a material handling system in which the components are consecutively assembled into a final product. A workstation is a physical area where a worker with tools and machines, or an unattended machine like robot performs a particular set of tasks. The components are processed depending on a set of tasks and they are performed at each workstation during a fixed time called as cycle time. The ALBP is to assign a set to tasks to workstation according to given precedence relationships among tasks with specific restrictions which aim to optimize one or more objectives, such as minimizing the number of stations and minimizing the cycle time. ALB occurs whenever assembly line is reconfigure, redesigned or adjusted to match the demand of new product or capacity. The main focus of ALBP is to obtain a task sequence which is feasible, minimizes workstation and ensures minimum waiting times in the workstation so that the efficiency of the line is maximized.

The techniques in GA have powerful performance for combinatorial optimization problems, especially for sequencing problems such as Traveling salesman problem (TSP) flow– shop scheduling and so on. However, when apply GA to synthesis practical sequencing problems, the infeasible chromosomes are often produced during crossover and mutation operations. Therefore, keeping feasibility of chromosomes might be considered as important issue when applying GA. In this paper, a repairing method based on topological sort is integrated with GA in order to handle precedence constraints and generated only feasible solution during the evolutionary process.

ALB Problem:

The problem is to assign a set of tasks to workstation with some measure of performance to be optimized under the following restrictions: (i) each task is assigned to one and only one workstation, (ii) the precedence relationship among the tasks cannot be violated, and (iii) the sum of the task times of any workstation should not exceed the cycle time. Since the task times allotted to workstations may be unequal, parts are produced at different speeds on the line. Accordingly, stations may either be starved or a queue may build up in front of a station. To regulate the flow of parts, assembly lines are often paced. In a paced line, each workstation is given a fixed amount of time called cycle time. If workstation finishes in less than cycle time given, it is to wait for the remaining period. The difference between the time required by any station to complete its operations and the cycle time is called the waiting time of the station. It is conventional to take the sum of all station waiting times (called total waiting time) as a measure of the efficiency of the design of a line.

The ALB is usually presented by the precedence graph. Consider a precedence graph in figure which specifies the order or sequence in which the task must be performed. The number in each circle refers task number, and the number above the circle refers the duration of the task. The arrow represents directions of flow of operation.

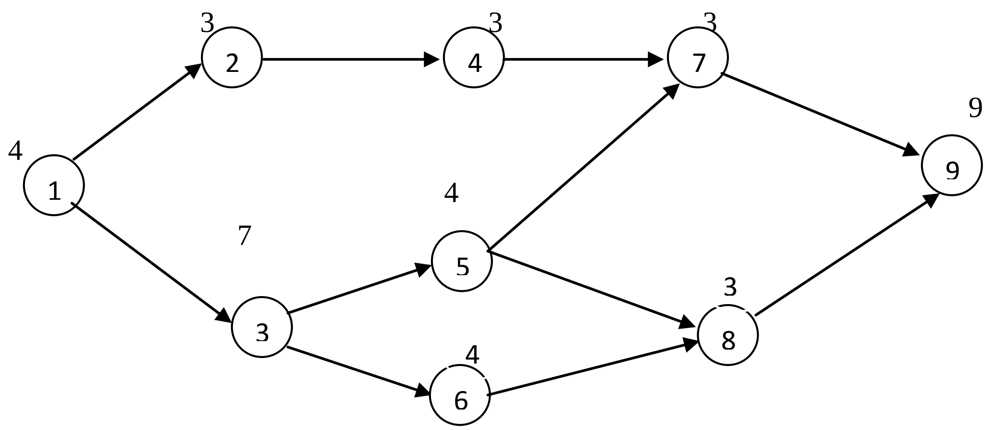

Figure-1 The precedence graph of simple assembly line.

The variable of interest for the ALBP consists of number of tasks (n), processing time, precedence relationships, and the cycle time (CT). The goals of the ALBPs are to minimize the number of workstations (m), minimize the waiting time (T_w) and maximize the line efficiency (E). Salveson was the first to give a mathematical form to the problem and proposed a LPP model to solve it. Formulations of workstation, waiting time, and line efficiency are given in (1) to (3) respectively, where 'w' is the total processing time and t_i is the processing time of the i^{th} workstation.

$$m = w / CT$$

$$T_w = \sum_{i=1}^{m} (CT - t_i)$$

$$E = \left(\sum_{i=1}^{m} t_i \right) / \left(mCT \right)$$

Topological Sort Repair method for feasible solution with GA.

The procedure of the repair mechanism using topological sort is given below.

Step 1 :

Form an initial available set of nodes (Events) having no predecessors, create an empty string.

Step 2:

Terminate, if the available set is empty. Otherwise, go to step 3.

Step 3:

Select a node (Event) from the available set at random, and append in to the string.

Step 4:

Update the available set by removing the selected node (Event) and by adding every immediate successor of the node if all the immediate predecessors of the successor are already in the string. Go to step. 2

In GA the infeasible Chromosomes in the initial population as well as the child's chromosomes created from the reproduction process need to be repaired before going through the evaluation process. Consider Figure and infeasible chromosome (1-6-5-3-2-4-7-9-8)

Table – I

Available Nodes	Infeasible chromosome	Updated sequence
{1}	**1** 3 5 6 2 4 7 9 8	1
{2,3}	1 3 5 **6** 2 4 7 9 8	1 6
{2,5,6}	1 3 5 6 **2** 4 7 9 8	1 6 2
{4,5,6}	1 3 **5** 6 2 4 7 9 8	1 6 2 5
{4,6}	1 3 5 6 2 **4** 7 9 8	1 6 2 5 4
{6,7}	1 3 5 6 2 4 **7** 9 8	1 6 2 5 4 7
{6}	1 **3** 5 6 2 4 7 9 8	1 6 2 5 4 7 3
{8}	1 3 5 6 2 4 7 9 **8**	1 6 2 5 4 7 3 8
{9}	1 3 5 6 2 4 7 **9** 8	1 6 2 5 4 7 3 8 9

Fitness Evaluation function

Fitness evaluation is to check the solution value of the objective function subject to the problem constraints. In this case, the objectives function is simply to minimize the total wait time.

$$\text{Minimize } T_w = \sum_{i=1}^{m} (CT - t_i)$$

Subject to precedence constraint.

Genetic parameters:

One of the main difficulties in building a practical GA is in choosing suitable values for parameters such as population size, crossover rate and mutation rate. The selections of parameter values are very depend on the problem to be solved.

Population size:

The population size is the number of candidate solution in any one generation. In principle, the population size should be sufficiently large such that the population associated with the solution space can be adequately represented. A larger population, however, needs larger computation cost in terms of memory requirement and computation time. The population size in this study is set to approximately half the total number of tasks.

Cross over rate Pc -

Crossover probability or crossover rate is how often will be crossover performed. If there is no crossover, child is exact copy of parents, but this does not mean that new generation consists of the same old individual. If there is a crossover, child is made from parts of parents chromosome. Most GA literature suggests that crossover rate should be set between 0.5 and 1.0.

Mutation rate, Pm -

Mutation rate is how often will be parts of chromosome mutated. If there is no mutation, child is taken after crossover or copy without any change. If mutation is performed, part of chromosome is changed. Mutation rate is often small which is the range 0.001 to 0.1.

Stopping condition:

Since GA is a stochastic search model it is quite difficult to formally specify a convergence criterion, as it is often observed that the fitness of a population may remain static for a number of generations before a superior string is found. If the GA has been correctly implemented, the population will evolve over successive generations so that the fitness of the best and average chromosome in each generation decreases / increases towards the global optimum. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached. In this study, numerical simulation experiments conducted based on a fixed maximum number of generations.

Generational gap:

It is possible that the chromosome with the highest fitness value in a generation may not survive selection process. A parameter called the generation gap was defined to control the fraction of the population to be replaced in each generation. Therefore 10% of the best fitness values in the population is kept and preserve for the crossover process.

Experimental Results

We present computational experiment results to evaluate the effectiveness of our proposed approach. For all experiments, the GA procedure utilize roulette wheel selection, Linear order Crossover (LOX) and inversion mutation to generate new candidate solutions in every generation. For each experiment, 10 trials of simulation are carried out and the minimum wait time is taken as an optimal solution.

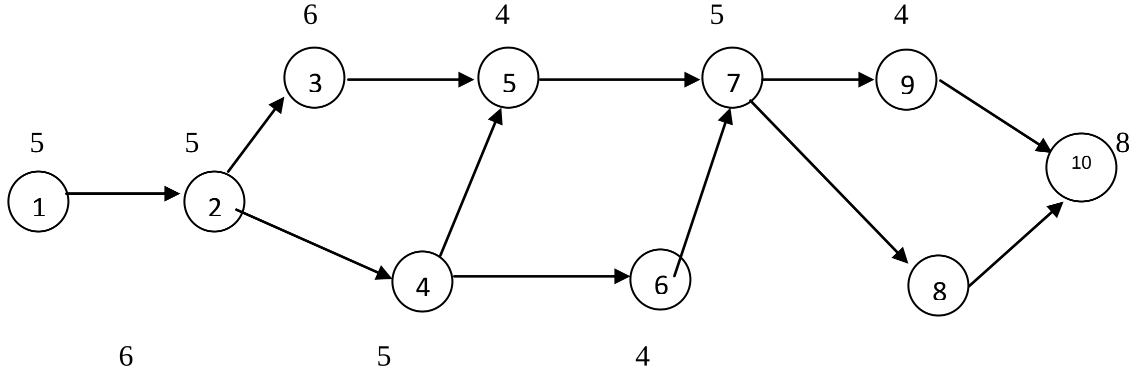

Figure – II Precedence graph of 10 tasks

Table – II

Optimal task sequence 1, 2, 4, 3, 5, 6, 7, 9, 8, 10			
Station	**Assigned Task**	**Processing Time (min) (t_i)**	**Waiting time (min) CT-t_i**
1	1, 2	10	0
2	4	6	4
3	3, 5	10	0
4	6, 7	10	0
5	9, 8	8	2
6	10	8	2

$\sum t_i = 52$

Total wait time (T_w) = 8

The line efficiency, E = 87%.

The proposed GA was tested on a single model ALB problems. The first experiment involve a simple ALBP which consists of 10 tasks and 10 precedence constraints as shown in figure - II. The total task time and the predetermined cycle time are 60 min and 10 min, respectively. The parameter setting of the GA which is Pc and Pm is 0.6 and 0.1 respectively. The population size is set to be half of the instance size and maximum number of generation is set as a termination criteria. Table shows the optimal solution obtained from the proposed GA in which the minimum waiting time and number of stations are 8 min and 6 station.

CONCLUSION

In this paper we discussed about the genetic algorithm combined with the topological sort procedure which is used to solve the ALBP. In this problem the total waiting time in the workstation is minimized. The topological sort is used for generating feasible task sequence while the genetic algorithm is for improving the quality of the solution. The result of the given problem shows that our approach is good interms of the quality of the solution. The presented GA approach is highly recommendable for practical applications.

Our study is only on a simple model of an assembly line with a single objective optimization. This can be developed for solving ALBPs with multi-objective optimization.

REFERENCES

[1] M.E. Salveson, “The assembly line balancing Problem”, Journal of Industrial Engineering, 6, 1955, pp.18-25.

[2] E.H.Bowman, “Assembly line balancing by linear Programming” Operations Research, 8(3), 1960, pp.385-389.

[3] D.E.Goldberg and K.Deb, A Comparative analysis of selection schemes used in genetic algorithms, in: G.J.E.Rawlins (Ed.), Foundations of Genetic Algorithms, Morgan Kaufmann, Los Altos, 1991, pp.69-93. Hdgjhd.

[4] A. Scholl, “Data of assembly line balancing problems” TH Darmstadt, http://www.assembly –line–balancing.de/.

[5] T.Ramaraj and G.Nirmala "Genetic Algorithm Matrix Neural Network Stochastic relaxation methods" Applied Science Periodicals, 2004.

[6] T.Ramaraj and G.Nirmala "Genetic Algorithm for Maximum model discrimination design – Matrix – Represented Solutions" Science and Technology – 2004.

[7] Carlos A. Coello Coello, Gary B.Lamont and David A. Van Veldhuizen "Evolutionary Algorithms for Solving Multi objective Problems" ISBN 978-81-8489-369-4.

[8] G.Nirmala and D.Ramprasad "Genetic Algorithm Approach for minimizing waiting times in Railway Networks" Narosa Publication (2010) ISBN 978-81-8487-074-9.

[9] G.Nirmala and D.Ramprasad "Genetic Operators with Algebra for equivalence relations" International conference on Algebra and its Applications PMU Journal, 2011.

[10] G.Nirmala and D. Ramprasad "Application of Genetic Algorithm in Graph Theory" International Journal of Scientific and Research Publications, Volume 2, Issue (July 2012), ISSN 2250 – 3153.