



# A PERMUTATION ENCODING TECHNIQUE APPLIED TO GENETIC ALGORITHM SOLUTION OF RESOURCE CONSTRAINED PROJECT SCHEDULING PROBLEM

M. H Oladeinde<sup>1</sup> and C. A Oladeinde<sup>2</sup>

<sup>1,2</sup>DEPARTMENT OF PRODUCTION ENGINEERING, FACULTY OF ENGINEERING, UNIVERSITY OF BENIN, NIGERIA.

*E-mail Addresses:*<sup>1</sup> [moladeinde@uniben.edu](mailto:moladeinde@uniben.edu), <sup>2</sup> [kallyama@yahoo.com](mailto:kallyama@yahoo.com)

## ABSTRACT

*In this paper, a permutation chromosome encoding scheme is proposed for obtaining solution to resource constrained project scheduling problem. The proposed chromosome coding method is applied to Genetic algorithm procedure and implemented through object oriented programming. The method is applied to a numerical example with two different stopping criteria. The results obtained are encouraging as they are in agreement with published solutions in literature. The method lends itself easily to programming and can be used to obtain results to large scale problems encountered in practice.*

**Keywords:** Genetic algorithm, resource constrained, project scheduling, makespan.

## 1. INTRODUCTION

Scheduling has received the attention of a large number of researchers. Among the scheduling problems addressed, project scheduling has been the most prominent. Traditional scheduling methods such as PERT (The Program Evaluation and Review Technique), [1, 2] and CPM (Critical Path Method), are not enough for Production scheduling because they assume that resource availability is unlimited. This assumption is not valid in most practical situations, in which there exist definite limit on the amount of resources available and these resources (renewable or nonrenewable) are shared by a number of activities or even projects. Resource constrained project scheduling problems (RCPSPs) involve assigning jobs or tasks to a resource or set of resources with limited capacity in order to meet some predefined objective. Different objectives are possible and these depend on the goals of the decision maker, but the most common of these is to find the minimum makespan [3]. The Resource-Constrained project scheduling problem (RCPSP) is the optimization problem in which the objective is subject to precedence relation between the activities and the limited renewable resource availabilities [4].

Since the pioneering work of Kelley [5], the resource-constrained project scheduling problem has been addressed by a great number of researchers [4]. Several exact procedures have been developed to solve the RCPSCP by a number of authors. An exact solution for the non-preemptive resource constrained project scheduling problem was formulated by [6]. It is impossible to obtain optimal solution to large-scale RCPSP problems because are known to be NP- Hard [7, 8]. A problem is classified as NP-hard if there are no known algorithms for finding optimal solutions in polynomial time. Hence to solve problem of the size generally experienced in practice, most research efforts focus on the development of efficient methods which produce good solutions but not necessarily optimal. These approximate methods have been used by authors include but are not limited to Genetic algorithm [9, 10], Ant colony optimization [11], simulated annealing [12] and Tabu Search [13]. Genetic algorithm mimics evolution in nature to find good solutions to problems. It uses the fact that only the fit survive. The chromosomes (father and mother) engage in crossover operation to produce other chromosomes (son and daughter). The chromosomes produced also mutate themselves. Due to these operators (crossover and mutation) fitter

chromosomes (individuals) are created as evolution occurs over generations.

The first step in genetic algorithm is to adopt a coding scheme for representing the genes in an individual. Encoding of chromosome depends on two factors namely order and value [14]. When fitness of the chromosome depends on order only, permutation encoding is ideal. Permutation encoding is used in ordering problems, such as traveling salesman problem or task ordering problem. When fitness depends on order and value, binary encoding scheme is used. In binary encoding, each object variable is encoded in a binary string (0 or 1) of a particular length defined by the user [15]. In value encoding, each chromosome is represented as the string of some value. The value can be integer, real number, character or some object. Binary Encoding is the most common representation of chromosome in genetic algorithms [16]. However, not all problems lend themselves to coding using the binary or permutation scheme. Specifically, in resource constrained project scheduling, a chromosome is represented by a schedule which is a permutation of the activities in a project. In this case, it is impractical to represent the chromosome using binary notation. In this paper, we propose a permutation encoding technique for solving resource scheduling in project networks under finite resources using genetic algorithm. The proposed method is illustrated using a numerical example for a single resource problem. Problems with multiple resources will be the subject of subsequent articles.

**2 MATHEMATICAL MODEL**

The resource constrained project scheduling problem consists of  $n+2$  activities which must be executed before the project can be completed. The first and last activities 0 and  $n+1$  are dummies and hence do not consume any resource and do not also take any time to execute. Let  $K = \{1, \dots, k\}$  be the set of resources. During the period  $d_j$  when an activity  $j$  is being processed,  $r_{jk}$  units of resource type  $k$  are required. The total amount of resource type  $k$  available at any instant is fixed and limited to  $R_k$ . The Resource Constrained Project Scheduling Problem (RCPSp) can be defined as finding a schedule of activities which minimizes the time it takes to complete the processing of all the jobs taking into account the precedence constraints as well as the limited availability of resource. Let  $F_j$  denote the finish time of activity  $j$ . A

schedule is given by a vector of finish times  $F_1, F_2, \dots, F_n$ .

Let  $A(t) = \{j \in J, F_j - d_j \leq t \leq F_j\}$  be the set of activities which are being processed at time  $t$

The linear programming model for the RCPSp as given by [17] is;

$$\text{Min } F_{n+1} \tag{1}$$

Subject to

$$F_h \leq F_j - d_j \quad j=1, \dots, n+1 ; h \in P \tag{2}$$

$$\sum_{j \in A(t)} r_{j,k} \leq R_k \quad k \in K ; t \geq 0 \tag{3}$$

$$F_j \geq 0 \quad j=1, \dots, n+1 \tag{4}$$

The objective function ensures that the finish time of the last activity is minimized. The constraint stated in Eq. (2) ensures that the precedence relations among the activities are satisfied while Eq. (3) ensures that the resource being utilized at any moment does not exceed the maximum capacity available.

**3. GENETIC ALGORITHM PROCEDURE**

Genetic algorithm is an evolutionary process used to obtain approximate solutions to problems. It mimics evolution in Genetics. A flow chart of the Genetic Algorithm process is shown in Figure 1.

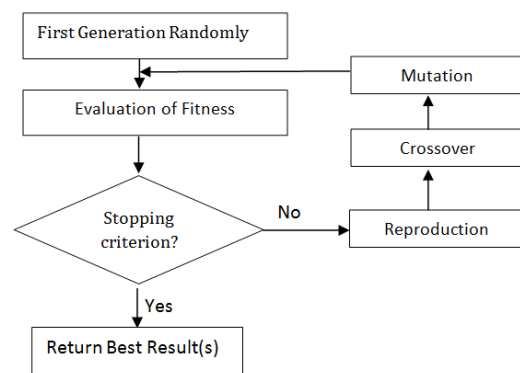


Figure 1: Genetic Algorithm flow chart

In the following sections, a description of the steps followed in the paper is presented.

**4. ENCODING OF CHROMOSOME**

In this work, the chromosomes are coded as a sequence (permutation) of project activity ID's. The strings indicate the activities in the project. For example in the project depicted by the network in Figure 2, a possible chromosome is shown in Figure 3.

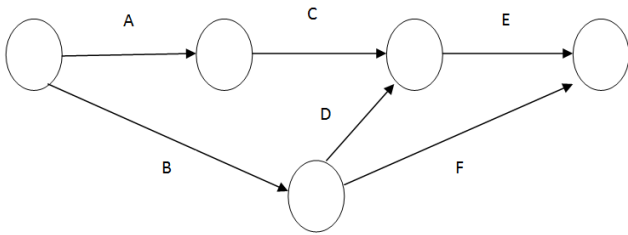


Figure 2: Network diagram of a project network



Figure 3: A chromosome for project network in Figure 2

We note that the arrangement of the strings in the chromosomes does not violate precedence constraints. The makespan of the schedule can then be found by drawing a Gantt chart (or finding the maximum of the vector of finish times) with resource on the ordinate and time on the abscissa, noting that in addition to precedence constraints, the resources required to execute an activity must be available before execution of the activity in question can commence and must be available throughout the execution of the activity (no preemption).

**5. INITIAL POPULATION**

The initial population comprises of a number of precedence feasible schedules generated by using a Random key based Serial generation scheme. The Random key serial generation scheme ensures that all the chromosomes of the initial population are precedence feasible. The randomness of the initial population generation scheme ensures that all potential precedence feasible chromosomes (schedules) have equal opportunity of being selected.

**6. COMPUTATION OF CHROMOSOME FITNESS**

The fitness of each schedule (chromosome) is calculated by finding the makespan of the chromosome given the resource constraint of the project. The makespan of a chromosome is obtained by developing a vector of finish times of the activities in the chromosome given resource constraints. The maximum finish time of all the activities in the chromosome is the makespan of the chromosome in question. Since the objective in resource constrained project scheduling is the minimization of the makespan, a chromosome with a small makespan is regarded as being fitter compared to one with a larger makespan.

**7. SELECTION**

In this operation, individuals which are fit are copied to the next generation. In this paper, the elitist selection criterion is used for choosing individuals in the current population to be copied to the next generation. Consider a generation with n chromosomes and the makespan of each chromosome is  $f_i$ . For each chromosome in the population, we determine  $f'_i$

$$f'_i = \frac{f_i}{\sum_{i=1}^{j=n} f_i} \tag{5}$$

The value of  $f'_i$  for each chromosome in the population is then arranged in decreasing order. The chromosomes which are the fittest (4 largest values of  $f'_i$ ) are copied to the next generation.

**8. CROSSOVER**

In this operation two individuals are selected as candidates for crossover. In crossover, the chromosomes exchange genetic material and hopefully will result in fitter individuals. There are seven forms of crossover, namely one point crossover, two point crossover, uniform crossover, half uniform crossover, reduced surrogate crossover, shuffle crossover and segmented crossover. In this work, the one point crossover was used and will be explained in the following section. Reference [19] may be consulted for details of the other forms of crossover.

**9. ONE POINT CROSSOVER**

In one point crossover, two chromosomes are randomly selected. One of the chromosomes is regarded as the Father and the other the mother. Consider two chromosomes which have been selected for crossover.

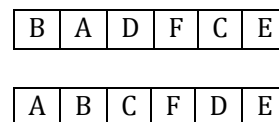


Figure 4: Two chromosomes selected for crossover.

The first chromosome is considered as the father and the second the mother. There are six genes (activities represented as strings) in the chromosomes. A random number is generated between 1 and n-1. Where n is the number of genes in the chromosome. In our example n = 6. This random number becomes the

crossover point. For example, assume the random number generated equal 2. The crossover operation results in two new chromosomes which we call the son and daughter chromosomes. The first three genes  $n = 1, 2$  are selected from the father chromosome and copied into the first three positions in the son chromosomes. The third to the sixth  $n = 3, 4, 5, 6$  genes are copied from the mother chromosome to form the son chromosome as shown in Figure 5.

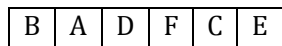


Figure 5: Son chromosome resulting from crossover operation with crossover point equal 2

In the case of the daughter chromosome, the first two genes of the daughter chromosome are selected from the mother chromosome and the others are selected from the father chromosome. The daughter chromosome is depicted in Figure 6.

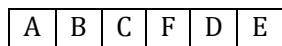


Figure 6: Daughter chromosome resulting from crossover operation with crossover point equal 2

The one point crossover operation is relatively simple for binary coding of chromosomes. However for resource constrained project scheduling problems, there is a restriction that the sequence of the strings resulting from the crossover operation must form a precedence feasible schedule (chromosome).

**10. MUTATION OPERATOR**

Mutation is an operation involving only one chromosome (unary). The mutation operation alters the chromosome by swapping two genes in the chromosome. When the two gene positions are swapped, it results in a new chromosome.

In string coding of chromosomes for resource constrained project scheduling however, the mutation operation is a little more challenging. This is because the mutation operation must result in precedence feasible chromosome.

**11. STOPPING CRITERION**

The Genetic Algorithm is essentially an iterative process in which we seek the optimal solution to a problem. As a result, we should be able to specify the conditions that should be met before the iterative process is discontinued and the best solution through all the generations taken as the optimal solution to the

problem. There are two ways that can be used to terminate the iterative process. One popularly used method is to terminate the search process when over a number of generations; there is no improvement in the solution. Alternatively, the user can specify a maximum number of generations after which the search is terminated. In this project, the second mode of termination of the search process is used.

**12. NUMERICAL EXAMPLE I**

In order to demonstrate the use of Genetic algorithm to the solution of Resource Constrained Project Scheduling problems two problems whose solutions have been obtained using other solution techniques are solved using the present method. The parameters used for obtaining the approximate solution using the Random key Genetic Algorithm solution were obtained from [18]. The network, as well as the activity durations, resource requirements for each activity and the resource constraints are presented.

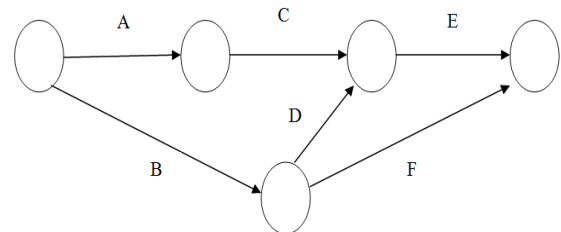


Figure 7: Arrow diagram of Project Network (Udosen, 1997)

Figure 7 shows the network diagram drawn with activity on arrow convention. The activity durations as well as the resource requirements are shown in Table 1. The resource constraint for the project network is equal to 2100 units per week.

Table 1: Activity durations and resource requirements for project network

Activity	A	B	C	D	E	F
Time (weeks)	3	2	4	3	2	1
Resource/week	1200	1000	700	1400	2100	200

The first step in obtaining a Genetic Algorithm solution to the Resource Constrained Project Scheduling Problem is to represent the project network using the Activity On Node (AON) convention. Dummy activities are introduced at the beginning and at the end of the project in order to convert the project network to the AON convention. The project network drawn using the AON convention is shown in Figure 8.

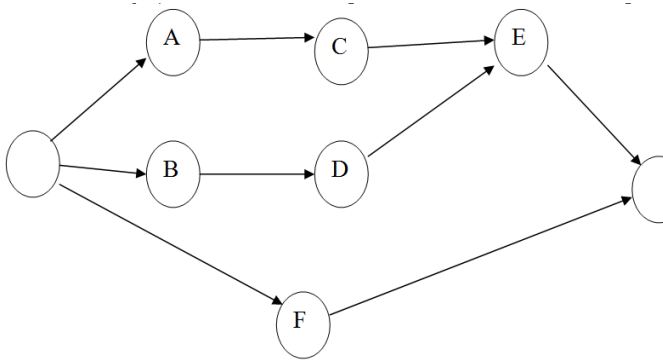


Figure 8: Project network using AON Convention

13. NUMERICAL RESULTS

The Genetic Algorithm is implemented using a Visual Basic.Net Program developed by the authors. The inputs to the program include the number of generations over which the search procedure is to be carried out. The program automatically generates the initial population which consists of eight precedence feasible schedules. The various steps listed in previous sections are carried out and the final solution displayed.

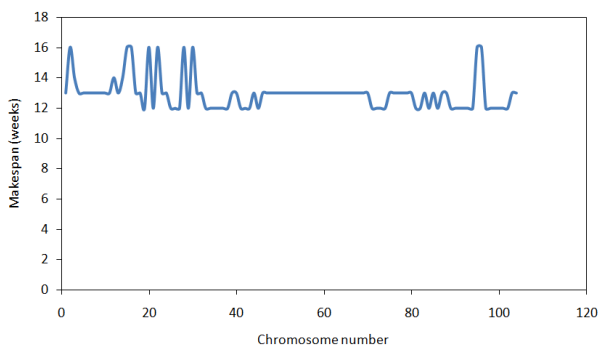


Figure 9: Solution obtained using GA for stopping criteria of 13 generations

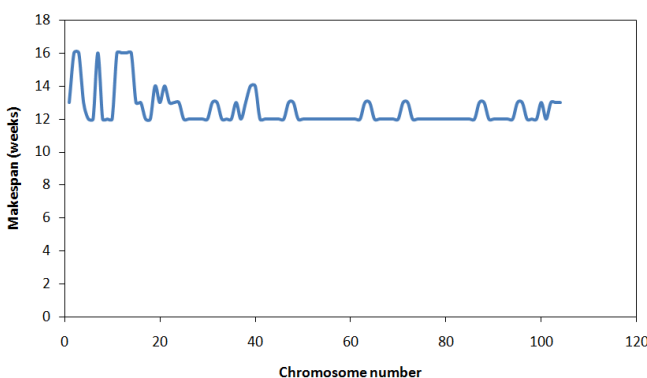


Figure 10: Solution obtained using Genetic Algorithm for stopping criteria of 40 generations

14. DISCUSSION OF RESULTS

The result obtained using Genetic Algorithm for the problem solved by [16] is shown in Figures 9 and 10. The initial population generated consists of eight

precedence feasible chromosomes. The iteration was first carried out for thirteen generations and the makespan of each chromosome in each generation plotted against the chromosome number. It is instructive to note that the makespan of each chromosome in each generation have been plotted in the ordinate while the chromosome number has been plotted in the abscissa. This has been done in order to retain complete information about the quality of the solution represented by the chromosomes in each generation. Figure 9 shows that over the first two generations, the least makespan obtained equal to 13 weeks. However, with increase in generation number, fitter individuals having makespan of 12 weeks are obtained. In the 7<sup>th</sup> generation, the makespan of all the chromosomes equal 13 weeks. Even though the makespan of all the chromosomes equal 13 weeks in the 7<sup>th</sup> generation, the genetic algorithm is able to produce fitter individuals with makespan of 12 weeks subsequently. Over the thirteen generations used as the stopping criterion, it can be seen that the least makespan of any chromosome equal 12 weeks. In order to validate the finding in figure 9 that the least makespan possible is 12 weeks, the program is ran with a stopping criterion of 40 generations. The progression of the makespan as the generation number increased is shown in Figure 10. The plot shows that the minimum makespan is equal to 12 weeks. This finding corresponds with that obtained by [16] who solved the same problem using the Tabular Graphical technique.

15. CONCLUSION

A genetic algorithm has been developed for resource constrained project scheduling problem using permutation encoding scheme for chromosomes. The method has been applied to a numerical example and the result obtained is in agreement with that published in literature.

16. REFERENCES

- [1] Anazodo, U. "System Approach to Farm Mechanization in Nigeria", *Nigerian Journal of Technology*, Vol. 1, No. 1, 1975, 7 – 11.
- [2] Oladeinde, M. H and Oladeinde, C. A. "An investigation into the Decision Maker's Risk Attitude Ranking", *Nigerian Journal Journal of Technology*, Vol. 33, No. 3, 2014, pp. 345 – 350.
- [3] Geunes, B. and Brien, W. "Resource constrained Project scheduling past work and new Directions",

- Research Report, Department of Industrial and Systems Engineering, University of Florida, Gainesville, 2001.
- [4] Okada, I., Zhang, X.F., Yang H.Y. and Fuzimura, S. "A Random Key-based Genetic Algorithm Approach for Resource – constrained project scheduling problem with Multiple modes", *Proceedings of the International Multi conference of Engineers and computer Scientists*, Vol. 1, 2010
- [5] Kelley, J., "The critical path method: Resource planning and scheduling," *Industrial Scheduling*, 1963, pp. 347-365.
- [6] Erenguc, S., Ahn, T. and Conway, D.G "The resource constrained project scheduling problem with multiple crushable modes: An exact solution method", *Naval Research Logistics*, vol 48, issue 2, 2001, pp 107 – 127.
- [7] Madan, N. and Madan, R. (2013), GASolver- A solution to Resource Constrained Project Scheduling by Genetic Algorithm, *International Journal of Advanced Computer Science and Applications*, vol 4, number 2, pp 210 – 217.
- [8] Blazewicz, J., "Complexity of computer scheduling algorithms under resource constraints," *Proc. 1st meeting AFCET-SMF on Applied Mathematics*, 1978, pp. 169-178
- [9] Alcaraz, J., Moroto, C., and Ruiz, R., "Solving the Multi-Mode Resource-Constrained project Scheduling problem with genetic Algorithms", *The journal of Operational Research Society*, vol 54, number 6, 2003, pp 616-626
- [10] Sakalanskas, L. and Felinskas, G., "Optimization of Resource Constrained Schedules by Genetic Algorithm based on the Job Priority List", *Information Technology and Control*, vol.35, number.4, 2006, pp 412-418
- [11] Merkle, D., Middendorf, M. and Schmeck, H. "Ant Colony Optimization for Resource-Constrained Project Scheduling", *IEEE Transactions On Evolutionary Computation*, vol. 6, no. 4, 2002, pp 333-346.
- [12] Chen, P.-H and Shahandasti, S.M., "Simulated Annealing Algorithm for Optimizing Multi-Project Linear Scheduling with multiple linear constraints" *24<sup>th</sup> International Symposium on Automation and Robotics*, 2007, pp 429-424
- [13] Skowronski, M.E., Myszkowski, P.B., Adamski, M., and Kwiatek, P., "Tabu Search approach for Multi-Skill Resource-Constrained Project Scheduling Problem" *Proceedings of the 2013 Federated Conference on Computer Science and Information Systems*, 2013, pp. 201153–158
- [14] Kumar, A. "Encoding Schemes in Genetic Algorithm", *International Journal of Advanced Research in IT and Engineering*, Vol 2, number 3, 2013, pp 1 – 7
- [15] Deb, K "Handbook of Evolutionary computation", Oxford University Press, 1997.
- [16] Jyotishree, R.K. "Novel Encoding Scheme in Genetic Algorithms for Better Fitness" *International Journal of Engineering and Advanced Technology (IJEAT)*, vol 1, Issue 6, 2012, pp 214 – 218
- [17] Christofides, N., Alvarez-Valdés, R., and Tamarit, J.M."Problem scheduling with resource constraints: A branch and bound approach". *European Journal of Operational Research*, vol 29, 1987, pp. 262-273.
- [18] Udosen, J.U. *Worked examples in network analysis with applications of # chain equations*, Essen Classic Nigeria Company Akwa Ibom, Uyo, 1997
- [19] Picek, S., and Golub, M. "Comparison of a Crossover Operator in Binary Coded Genetic Algorithms", *WSEAS Transactions on Computers*, volume 9, issue 9, 2010, pp 1064-1073.