Optimization of the hydrotesting sequence in tank farm construction using an adaptive genetic algorithm with stochastic preferential logic

J Lancaster[1,2]* and **K Cheng**[1]
[1]Advanced Manufacturing and Enterprise Engineering, Brunel University, Uxbridge, UK
[2]Hill International, Abu Dhabi, United Arab Emirates

The manuscript was received on 11 October 2007 and was accepted after revision for publication on 30 October 2007.

DOI: 10.1243/09544054JEM1022

Abstract: In the construction of tank farms there is a requirement for the tanks to be hydrotested in order to verify that they are leak proof as well as proving the lack of differential settlement in the foundations. The tanks will be required to be filled to a predetermined level and then to maintain this loaded state for a certain period of time before being drained. In areas such as the Middle East, water for hydrotesting is not freely available as sea water is often not suitable for this purpose, so fresh water needs to be produced or transported to the construction site for this purpose. It is therefore of major benefit to the project to schedule the hydrotesting of the tanks in such a manner as to minimize the utilization of hydrotest water.

This problem is a special case of the resource-constrained project scheduling problem (RCPSP) and in this research the fitness differential adaptive genetic algorithm previously developed by the authors has been modified to enable the solution of this real world problem. The algorithm has been ported from the original MATLAB code into Microsoft Project using Microsoft Visual Basic for Applications in order to provide a more user friendly, practical interface.

Keywords: genetic algorithm, resource-constrained project scheduling problem (RCPSP), project scheduling

1 INTRODUCTION

This paper aims to provide a solution to a special case of the well-studied resource-constrained project scheduling problem (RCPSP) utilizing a genetic algorithm approach. Overviews of previous research in this area can be found in both Lancaster and Ozbayrak [**1**] and Kolisch and Hartmann [**2**]. The unique case considered in this paper is that the determination of certain preferential logic links will be considered as part of the optimization process. Certain activities are identified that may be considered for the application of preferential logic and a stochastic process is used to apply this logic. This additional logic is then stored as an extension to the normal genetic algorithm chromosome and is therefore refined as part of the optimization process. The problem of hydrotesting a number of tanks in the construction of a storage tank farm is considered for application of this technique, as this type of preferential logic is required in order to ascertain the sequence in which the hydrotest water will be transferred from tank to tank.

Oil refineries and import terminals normally require the construction of large tank storage farms. These farms comprise a large number of tanks of differing volumes and designs. One normal requirement on the construction of all of the tanks is that they be tested by partial filling with water in order to test for leakage and differential settlement in the foundations. Large volumes of water are required for the testing of these tanks and in order to reduce the total quantity employed, water is transferred from one tank to another, but with many differing capacities planning the sequence is complex, as to maintain a certain quantity of water within the system the total volume of tankage under hydrotesting at any one time will need to be approximately consistent. The objective of this optimization problem is to identify the

Corresponding author: Claims and Construction Management, Hill International, PO Box 5201, Abu Dhabi, United Arab Emirates. email: JohnLancaster@Hillintl.com

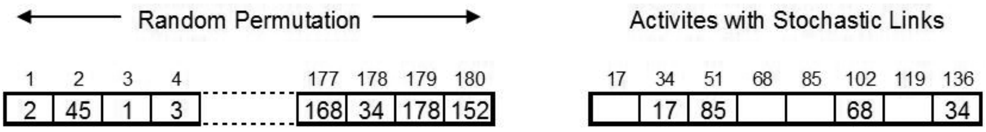

Fig. 1 Chromosome extension

sequence with the minimum duration that will utilize no more than the total available volume of water.

This optimization becomes particularly important in arid areas such as the Middle East where water is at a premium. Sea water is often not suitable for this testing purpose and therefore fresh water will often need to be produced or transported to the construction site for this purpose at a large expense.

The current paper is structured as follows: section 2 looks at the additional requirements for optimizing this problem over those of the standard RCPSP; section 3 discusses the structure of the current algorithm; section 4 outlines the test problem utilized to test the effectiveness of the algorithm; section 5 discusses the results and finally section 6 presents conclusions and suggestions for future research.

2 THE HYDROTESTING PROBLEM

On first consideration it would seem that the problem described would be a simple implementation of the RCPSP; the water used in hydrotesting can be modelled as a renewable or non-renewable resource depending on the approach adopted, and performing a levelling of this resource within certain limits, while minimizing the duration, would be a typical RCPSP problem and would indeed to a certain point provide optimization of this problem. However, the vital component that would be missing when optimizing this problem in the manner just described is that it would not provide, as output, a routing, or logical sequence in which water is transferred from one tank to another.

In order to include this additional requirement into the model, stochastic logic has been adopted into the problem. Items in the schedule to be considered for stochastic logic are marked prior to the optimization run. As part of the processing of each potential schedule, logical relationships are formed randomly between the marked activities. This stochastic logic is then considered along with the existing deterministic logic during the normal processing of the population of schedules.

3 THE GENETIC ALGORITHM

The main construction of this present algorithm is based on that presented in Lancaster and Cheng [**3**], which uses adaptation based on the fitness differential between successive generations to modify the mutation factor. The stochastic logic has been modelled as an extension to the existing chromosome, providing a position in the chromosome for each of the activities identified for stochastic logic.

Figure 1 shows the extension of the chromosome. In the test problem a 180-activity schedule is utilized with 8 activities selected for stochastic logic relationships. The random permutation portion of the chromosome shown in Fig. 1 holds the sequencing of all 180 activities. The extension portion shown to the right holds 8 additional alleles (the term for a single position within the chromosome) to map the randomly generated logic. The actual values contained in these 8 positions refer to the stochastic successor of that particular activity. This randomly generated logic is used during the schedule generation along with existing deterministic logic to produce the feasible schedule.

One consideration that needs to be made is that by purely applying random logic generation it is necessary to avoid the creation of logic loops that would prevent the formation of a feasible schedule. The present algorithm performs loop checking utilizing a depth first recursive cycle check in order to first identify loops where they exist and then to select a logic link to remove in order to break the loop.

The utilization of an extension of this nature to the normal permutation portion requires specialized cross-over and mutation operators that are capable of allowing for this functional division of the chromosome into two portions, performing their operations on the two portions separately. In order to cater for this a composite cross-over operator was developed. This cross-over operator consists of two components; the first component is a standard independent cross-over operator (IPX [**4**]), which is applied to the traditional part of the chromosome, i.e. the alleles containing the permutation of the activities or the 'activity list'. The second component, a two-point cross-over is utilized on the chromosome extension. This cross-over operator is illustrated in Fig. 2.

In addition to the specialized cross-over operator a complimentary mutation operator has also been designed. This mutation operator also comprises two components. The first component applied to the main chromosome is a two-point adjacent swap mutation operator (Murata and Ishibuchi [**4**]) and the second component is simply a single-bit random

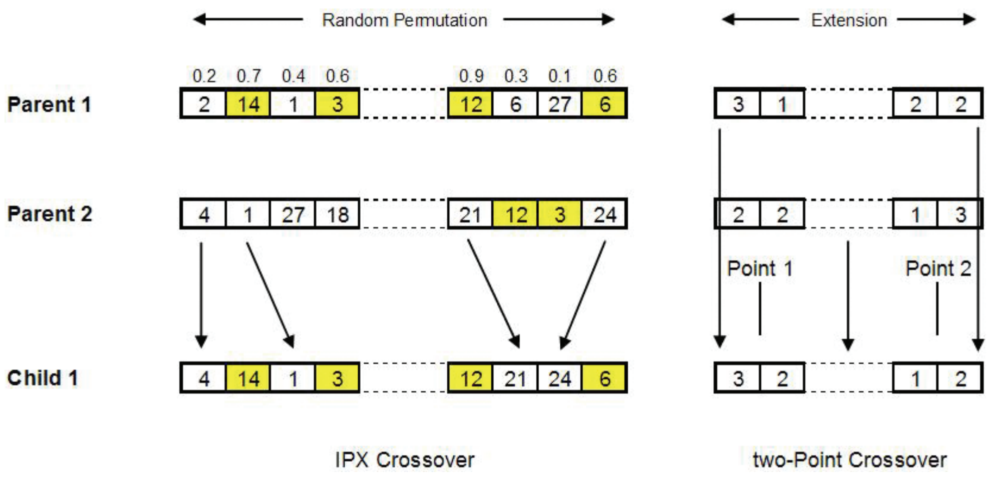

Fig. 2 Extended cross-over operator

change operator which is applied to the chromosome extension. This second mutation operator selects an allele from the chromosome extension at random and then changes its value to a randomly selected member of the set of activities identified for stochastic logic application.

Once the cross-over and mutation have been performed, a cycle checking algorithm is employed to ensure that loops have not been introduced to the network via the function of the genetic operators. This algorithm will also break any detected loops to ensure a feasible network remains. The loops are only broken by removing links that have been stochastically assigned; hence the integrity of the original network is always maintained.

In order to communicate the logic into the algorithm an adjacency matrix is utilized. The logic links contained in the adjacency matrix are considered in two sets: the first set is the deterministic logic links which remain constant for the entire optimization; the second set is the stochastic logic which will change for each chromosome considered. To manage this within the algorithm a copy of the deterministic adjacency matrix is made prior to applying the serial schedule generation scheme. The stochastic logic for the chromosome being considered is added to the deterministic logic and the schedule generation algorithm is then run. Figure 3 shows a sample of the adjacency matrix with the addition of stochastic logic carried out at the processing of each chromosome; the example only shows activities from the first tank with the hydrotest of activity of the second tank, a stochastic link between these hydrotests has been indicated.

The authors' original fitness differential adaptive genetic algorithm [**3**] has been adapted for this problem and has been rewritten from the original MATLAB code into Visual Basic for Applications (VBA) within Microsoft Project in order to provide a better platform for practical implementation of this algorithm.

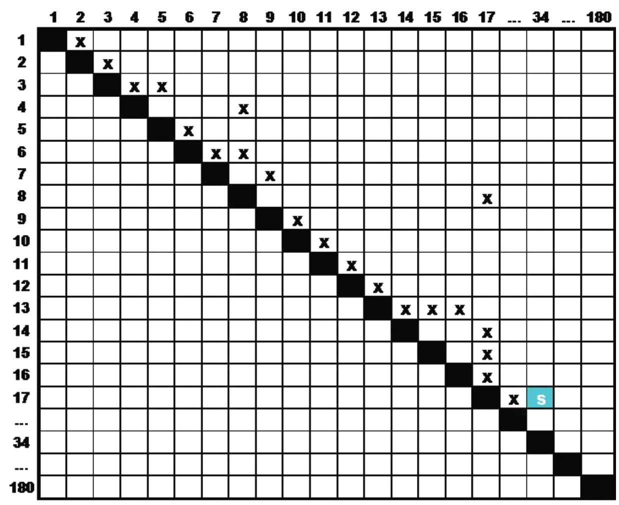

Fig. 3 Adjacency matrix addition

4 THE TEST PROBLEM

The test problem being utilized as an example for this problem is a 180-activity schedule representing the construction of an 8-tank tank farm. From these activities the 8 hydrotesting activities (1 for each tank) have been selected for stochastic logic assignment. The application of stochastic logic to this problem can result in between 1 and 8 transfer paths for the hydrotest water, 8 separate paths if no logic is applied, and 1 path if a continuous sequence is formed with logic being applied to all of the stochastic logic activities.

It is the aim of this problem to utilize the application of preferential logic in order to maintain the desired level of resource utilization while also determining a suitable logic path from the hydrotest of one tank to that of another. The target maximum resource utilization is 140 000 units of water, for the purpose of this problem the resource levels of other activities are ignored.

In the typical RCPSP, activities are scheduled at their first precedence and resource feasible time. In this problem only precedence feasibility is considered and the variation in preferential logic is utilized to provide the vehicle for optimization. The fitness function of the problem has been altered so that the algorithm will aim at minimizing the resource utilization. When a utilization is obtained that falls below the desired limit, a reward factor is applied to the fitness measure of that solution. Using this philosophy a logic sequence will be produced that will maintain the desired resource level. In this specific problem this will produce a transfer path for the hydrotest water from tank to tank that will minimize the total water usage.

5 RESULTS

The solution provided by the algorithm produces the Gantt chart and resource histogram that are shown in Fig. 4. The logical path for the hydrotest water through the 8 tanks can be seen from Fig. 4. Activity 34 which requires 2 353 000 units of water is predecessor to activity 68 also requiring 2 353 000 units and this sequence continues through activity 51 and 17. A similar chain is formed between the four larger tanks requiring 3 360 000 units. The two paths that the algorithm has selected are: 34–68–51–17 and 136–119–85–102. Figure 4 also shows that the prescribed resource limits have been met by the problem with the peak resource usage being 140 000 units.

6 CONCLUSIONS AND DIRECTIONS FOR FUTURE RESEARCH

This paper has described a genetic algorithm optimization of a project scheduling problem that utilizes preferential logic optimization in order to meet resource requirements. A practical problem has been described in order to provide evaluation of the effectiveness of the solution and the algorithm has successfully provided the desired results.

A specific problem faced by stochastically applying preferential logic was described – the formation of cycles within the network and the solution employed within the algorithm for solving this issue was also discussed. This method of optimization is a parallel of the often used manual process of resource levelling. By applying logic to selected activities it ensures that resources are levelled in a manner that

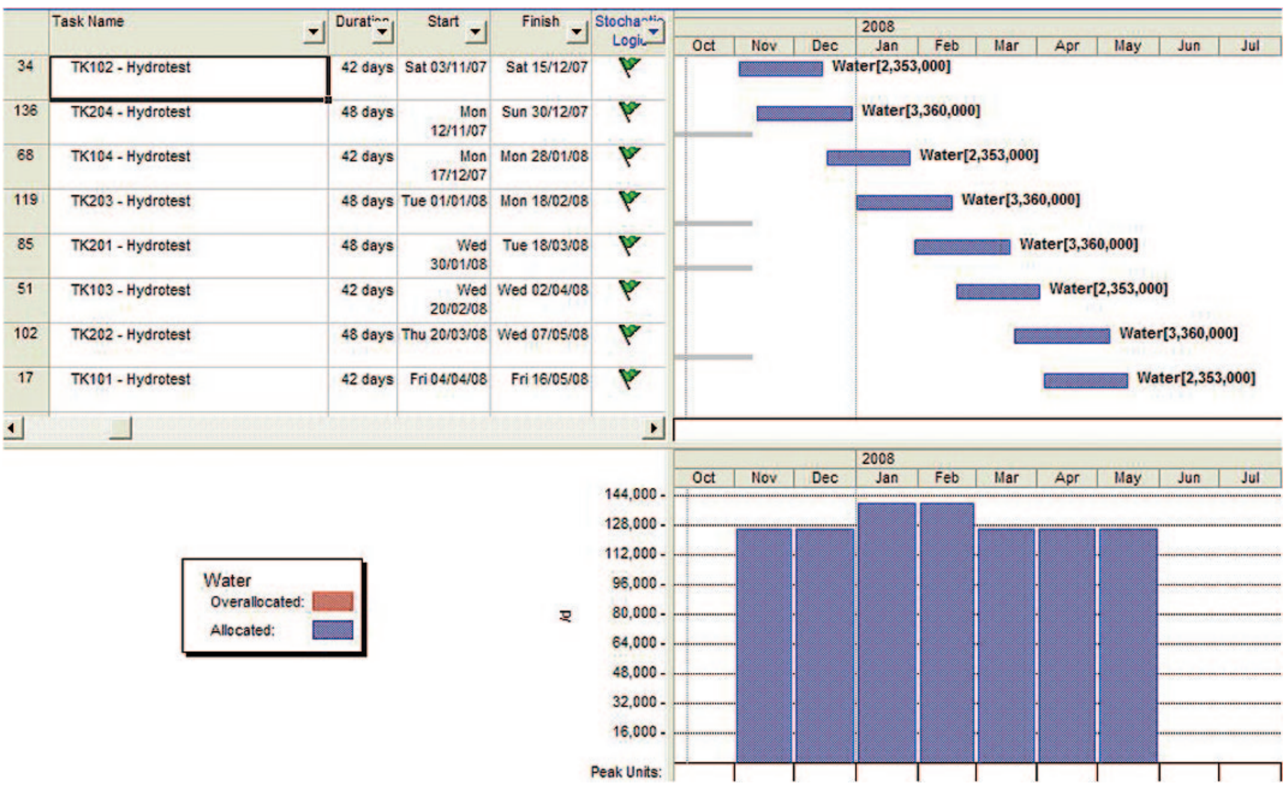

Fig. 4 Optimization of case study problem

will provide an executable schedule. Further work can be conducted in combining this technique with the standard RCPSP and also through multi-objective optimization where duration and resource minimization are considered.

REFERENCES

1 **Lancaster, J.** and **Ozbayrak, M.** Evolutionary algorithms applied to project scheduling problems: a survey of the state-of-the-art. *Int. J. Prod. Res.*, 2007, **45**(2), 425–450.

2 **Kolisch, R.** and **Hartmann, S.** Experimental investigation of heuristics for resource constrained project scheduling: an update. Available online at http://129.187.106.231/psplib/files/KH-18–2-05.pdf, last accessed 12 October 2007.

3 **Lancaster, J.** and **Cheng, K.** A fitness differential adaptive parameter controlled evolutionary algorithm with application to the design structure matrix. *Int. J. Prod. Res.* (in press) 10.1080/00207540701324176.

4 **Murata, T.** and **Ishibuchi, H.** Performance evaluation of genetic algorithms for flowshop scheduling problems. In Proceedings of the First IEEE Conference on *Evolutionary computation*, 1994, **2**, pp. 812–817.