A New Genetic Algorithm for the RCPSP in Large Scale

Hossein Zoulfaghari, Department of Industrial Engineering, University of Tabriz, Tabriz, Iran

Javad Nematian, Department of Industrial Engineering, University of Tabriz, Tabriz, Iran

Nader Mahmoudi, Department of Industrial Engineering, University of Tabriz, Tabriz, Iran

Mehdi Khodabandeh, Department of Industrial Engineering, University of Tabriz, Tabriz, Iran

ABSTRACT

The Resource Constrained Project Scheduling Problem (RCPSP) is a well-studied academic problem that has been shown to be well suited to optimization via Genetic Algorithms (GA). In this paper, a new method will be designed that would be able to solve RCPSP. This research area is very common in industry especially when a set of activities needs to be finished as soon as possible subject to two sets of constraints, precedence constraints and resource constraints. The presented algorithm in this paper is used to solve large scale RCPSP and improves solutions. Finally, for comparing, results are reported for the most famous classical problems that are taken from PSPLIB.

Keywords: *Genetic Algorithm, Large Scale Problems, Precedence Constraints, Resource Constraints, Resource Constrained Project Scheduling Problem (RCPSP)*

INTRODUCTION

The resource Constrained project scheduling problem (RCPSP) is a classical and changeable optimization problem that has attracted many researchers and a well-known problem where the activity of a project must be scheduling to minimize its makespan (Brucker et al., 1999; Herroelen et al., 1998; Kolisch & Hartmann, 1998). Not only exact solution procedures, but also many heuristic methods have been proposed to solve RCPSP. Lancasrer and Ozbayrak (2007) and Kolisch and Hartmann (2005) have provided detailed history of the work conducted in this area to the date.

Being an NP-hard problem, Alcaraz and Maroto (2001) mentioned that the optimal solution can be achieved by exact solution procedures but only in small projects, usually with less 60 activities. Heuristic methods are designed to solve large and high resource-constrained projects. Möhring et al., (2003) have mentioned that RCPSP is one of the most intractable problems in operations research; and many optimization techniques have applied to solve it. Many effective heuristic and meta-heuristic algorithms are also proposed for RCPSP. Several authors have suggested different methods to solve this problem. They can be classified into three categories, the exact methods (Brucker et al.,

DOI: 10.4018/jaec.2013040103

1998; Demeulemeester & Herroelen, 1992; Christofiedes et al., 1987; Mingozzi et al., 1998; Patterson, 1984) generated optimal solutions for small size problems. There are many exact algorithms proposed for RCPSP, which are mainly based on the branch-and-bound strategy. For example, Demeulemesster and Herroelon (1997) developed depth-first branching rules. Brucker et al. (1998) presented a branch and bound algorithm where its branching scheme is applied on a set of conjunctions of disjunctions to pair of activities. As an NP-hard problem, the optimal solution can only be achieved by exact methods in small projects, which are not highly resource constraints.

Heuristic procedures are the algorithms for large projects, the best ones gives good solutions with few computational requirements in a reasonable computation time .We can mention the general work of Brucker et al. (1998) as well as the work of Kolisch and Hartmann (1998) which focuses on heuristic algorithms to solve the problem. Due to limitations of exact algorithms, some authors proposed heuristic algorithms for RCPSP.For example, Möhring et al. (2003) proposed a heuristic based on the Lagrangian relaxation and minimum cut computations .The heuristic methods which are based on priority rules can be divided into two classes: Single-pass methods and multi-pass methods which were presented in Boctor, 1990. The forward-backward methods were also proposed in Li & Willis 1992.

The third category includes meta-heuristic methods like Tabu search (Baar et al., 1997; Pirson et al., 1994), simulated annealing (Boctor, 1996; Bouleimon & Lecocq, 2003) and genetic algorithms (Alcaraz & Maroto, 2001; Hartmann, 1998; Kohlmorgen et al., 1999). Recent numerical comparisons of these methods (Alcaraz & Maroto, 2001; Bouleimon & Lecocq, 2003; Hartmann & Kolisch, 2000) indicate that the three most efficient procedures are (in order) the genetic algorithms of Alcaraz and Maroto (2001), the simulated annealing methods of Bouleimen and Lecocq (2003) and the genetic algorithm of Hartmann (1998). To improve Hartmann genetic algorithm using an activity list encoding (representation) of the solution, Alcaraz and Maroto (2001) introduce an addition component in the encoding to indicate the scheduling mode (forward-backward) used to generate the corresponding schedules. The investigation of Hartmann and Kolisch (2004) and its update version (Hartmann & Kolisch, 2006) conducted an elaborate study on state-of-the art heuristic and meta-heuristic methods. They presented performance comparisons among heuristic and meta-heuristic methods in their study by applying these methods to different standard instance sets, namely J30, J60 and J120, generated by Pro-Gen in the PSPLIB. As shown by the latest experimental evaluation (Hartmann & Kolisch, 2006) meta-heuristic methods outperform heuristic methods. One of the first attempts to apply genetic algorithm (GA) in the scheduling problems was made by Davis (1985). The main idea of his approach was to encode the representation of schedule in a meaningful and legal way. Alcaraz and Maroto (2001) developed a GA based on the activity list representation and the serial SGC. An additional gene was used to decide whether forward or backward scheduling is employed when computing a schedule from an activity list. The crossover operation for activity list was extended such that a child's activity list could be built up either in forward or in backward direction. Alcaraz, Maroto, and Ruiz (2004) extended the genetic algorithm of Alcaraz and Maroto (2001) by adding two features from the literature. First, they took the additional gene that determines the SGC to be used form Hartmann (2002). Second, they employed the forward-backward improvement of Tormos and Lava (2001). Mendes et al. (2009) used a random key representation and a modified parallel SGS. The modified parallel SGS determined all activities to be eligible which could be started up to the schedule time plus a delay time. The random key had twice the length of the number of activities and each entry was a random number. The first half of the entries biased the activity selection and the second half biased the delay time of the SGS. They also presented a new genetic algorithm for multi-project scheduling

Related Content