STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVII**, Number 3, 2012

A COLLABORATIVE EVOLUTIONARY APPROACH TO RESOURCE-CONSTRAINED PROJECT SCHEDULING

ANCA ANDREICA, CAMELIA CHIRA

Abstract. Resource-constrained project scheduling is an NP-hard optimization problem focusing on the task of time-dependent resource allocation for a project. The current paper presents the application of a geometric collaborative evolutionary algorithm to this problem. Important features of the evolutionary model include population topology, asynchronous search and adaptive selection/recombination strategy. Each individual has an agent-inspired behaviour in the sense that communication with other individuals is possible and facilitates the selection of a mate for recombination. The evolving population has a geometrical structure and is furthermore organized in dynamic societies with different strategies for recombination. Numerical experiments are performed for several project instances and results emphasize a good performance of the geometric collaborative evolutionary model.

1. Introduction

The Resource-Constrained Project Scheduling Problem (RCPSP) is of great importance in a large number of application areas including construction and civil engineering, manufacturing, production planning, logistics and project management. RCPSP requires the allocation of limited resources to dependent activities over time, such that the makespan of the project is minimized. The challenges associated with RCPSP relate to complex resource constraints as well as activity dependencies. It has been shown that RCPSP belongs to the class of NP-hard optimization problems [1] which means that exact solutions can not be found in polynomial time by running an algorithm. Therefore, there is a high interest in developing good approximation methods to address RCPSP with the aim of finding near-optimal (or optimal) solutions using limited resources. Inspired by the process of natural evolution, genetic

Received by the editors: October 8, 2012.

2010 *Mathematics Subject Classification.* 68-02.

1998 *CR Categories and Descriptors.* I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search – *Heuristic methods*.

Key words and phrases. evolutionary computation, resource-constrained project scheduling, crossover, permutation based encoding, collaborative evolutionary algorithms.

algorithms represent good approximation methods for solving problems belonging to this class. Bio-inspired heuristic approaches to RCPSP include a genetic algorithm with a new permutation of priority-based encoding scheme [13], a permutation-based elitist genetic algorithm [8], an algorithm based on a priority value encoding scheme [12] and a hybrid genetic algorithm [11].

The evolutionary framework used in this paper is the Geometric Collaborative Evolutionary (GCE) model proposed in [2]. In this model, the population has a geometrical structure given by the circular placement of individuals on layers according to their fitness. Furthermore, the agent-inspired component of GCE leads to the collaboration of individuals which facilitates selection and recombination. The search process is asynchronous allowing the improvement and replacement of individuals within the same generation. The three main strategies for recombination are supported by three co-evolving and dynamic subpopulations or societies of individuals (i.e. *local*, *far* and *global*) having different policies for mate selection and recombination. Population dynamics emerges through the recombination of individuals from different societies and a dominance principle. Co-evolution of societies enables a useful balance between search diversification and intensification. Some individuals are specialized for local search facilitating exploitation while other individuals focus on global search. The GCE model reports promising results for various difficult unimodal and multimodal real-valued functions with many dimensions [2, 5] and for the problem of evolving Cellular Automata rules [3] in which the real and, respectively, binary representation have been used.

In the current paper, the GCE model is adapted for the application to RCPSP based on the permutation representation. Computational experiments are presented for RCPSP based on several project instances and the results of the RCPSP-customized GCE algorithm are compared to that of a standard evolutionary algorithm.

The paper is organized as follows: Section 2 presents the RCPSP addressed in this paper, Sectiom 3 describes the GCE model, Section 4 presents the GCE-based approach to RCPSP, Section 5 discusses the experimental results obtained and Section 6 contains the conclusions of the paper and some directions for further research.

2. The Problem of Resource-Constrained Project Scheduling

RCPSP [1] considers a project with a set of activities and a set of available resources. Let J be the number of activities (also called jobs) and $\{a_1, ..., a_J\}$ the set of activities. Jobs have to follow certain precedence constraints which means that a job can not start before all its predecessors are finished. Let us denote by P_j the set of all predecessors of activity a_j and by S_j the set of all its successors. The precedence constraints are usually represented as an acyclic activity-on-node network. Two additional activities are also normally considered: an initial activity a_0, which must precede all other activities of the project and a final activity a_{J+1} which must be preceded by all activities of the project.

In order to be executed, each activity requires a certain amount of some of the available resources. Let K be the set of existing resources. In this work, we only consider renewable resources - characterized by a constant per-period-availability (R_k, for each $k \in K$). The amount of resource k needed by the activity a_j is denoted by r_{jk}. Furthermore, each activity a_j has a fixed duration or processing time denoted by p_j. It should be noted that the dummy activities (a_0 and a_{J+1}) require no time and no resources.

A schedule assigns a start time to all activities $a_1, a_2..., a_J\}$ with the property that the end time of an activity a_j is the sum of its start time and duration p_j. This makes the project makespan to be equal to the start time of the final dummy activity a_{J+1}. A schedule is feasible if at any time the demand for resource k does not exceed its availability: $\sum_j r_{jk} \leq R_k$.

The goal of RCPSP is to find a schedule of the activities with minimum makespan taking into account the precedence and the resource constraints.

3. The Geometric Collaborative Evolutionary Model

The *GCE* model [2, 3, 5] integrates agent-based behavior into the evolution of the population in order to facilitate the adaptation of individuals to the environment as the search process progresses.

The population has a geometrical structure which allows the definition of a neighborhood notion used in designing different selection strategies (see Figure 1). Initially, all individuals are sorted according to their fitness and distributed over concentric layers starting with the most fit individuals on the most inner layers. The population size is fixed at n^2 (where n is an even number) which leads to a number of $n/2$ layers, each layer i ($i = 0, ..., n/2-1$) having $4(n - 2i - 1)$ individuals. Let us denote the sorted population at iteration t by $P(t) = (x_1, x_2, ..., x_{n^2})$, where x_1 is the fittest and x_{n^2} is the worst individual in the population. The most inner layer contains the first four individuals (x_1, x_2, x_3, x_4). The next layer holds 12 individuals $(x_5, ..., x_{16})$ having the next best fitness values. The most outer layer is labeled by 0 whereas the label of the most inner layer is $n/2 - 1$. The example depicted in Figure 1 uses a population of 8^2 individuals which leads to 28 individuals being placed on layer 0 down to the best 4 individuals placed on layer 3.

Based on this population topology, *local* selection refers to individuals situated on the immediately previous layer (better but still resembling fitness),

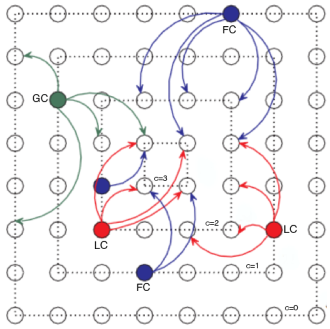

FIGURE 1. GCE Population Topolgy and Society Strategies.

far selection seeks individuals from more distant layers (at least two layers apart) containing more fit individuals while *global* selection considers the entire population. The agent-inspired component of the GCE model refers to the organization of the population in three societies of individuals co-evolving during the search process. Each individual can be viewed as an agent with the objective of optimising its fitness being able to communicate and select a mate for recombination. Individuals belong to one the following three societies: *Local Correlation (LC)*, *Far Correlation (FC)* and *Global Correlation (GC)*.

The GCE model uses an asynchronous search scheme. Individuals from a layer are updated through recombination and are involved in forthcoming recombination processes within the same generation. The individuals from the most inner layer are automatically copied in the next population (as an elitist strategy). Each individual from the population has the chance of being improved by getting involved in a recombination process.

Recombination in the GCE model is guided by the following strategies corresponding to each of the three agent societies (see Figure 1):

- LC individuals from a layer c address mating invitations to individuals from layer $(c + 1)$, where $c = 0, ..., n/2 - 2$.
- FC individuals from a layer c address mating invitations to individuals from layer $(c+i)$, where $c = 0, ..., n/2-3$ and $i \geq 2$ is randomly selected using a uniform distribution. FC individuals from layer $(n/2-2)$ invite individuals from layer $(n/2 - 1)$.
- GC individuals from a layer c may address mating invitations to individuals from any layer except layer c.

Furthermore, each individual invited to be a mate can accept or decline the proposal according to its own strategy. Normally, individuals from *LC* and *FC* societies accept individuals from the same society or from the *GC* society as mates. Individuals from *GC* society accept any other individual as mate. Offspring are assigned to a certain society according to a dominance concept. If LC is the dominant agent society then any combination of a GC individual with a LC individual results in an offspring belonging to LC.

A probability of one society dominating another is used to modulate the interactions between individuals belonging to different societies. Let p be the probability of LC (and FC) dominating GC. The dominance probability p may be viewed as the (probabilistic) membership degree of an offspring to the society LC (FC) when one of the parents is a GC individual. Several assignment schemes for p are analysed in [5] focusing on the dynamics given by the co-existence of the three societies of individuals.

For each mating pair (x, y) the offspring z obtained after recombination is mutated obtaining $mut(z)$. The best between z and $mut(z)$ is compared to the first parent x and replaces x if it has a better quality. The elitist scheme that allows only better individuals to replace the first parents is mitigated by the fact that all individuals from the population are involved in recombination.

The importance of the agent-inspired component of GCE has been investigated in [3] and the results support the hypothesis that an adaptive behavior of individuals within an evolving population benefits the search process. This adaptive behavior is triggered in the GCE model by the interactions between individuals belonging to societies with different strategies for selection and recombination.

4. Evolutionary Algorithm for RCPSP

The evolutionary approach to RCPSP developed in the current paper is an instantiation of the GCE model. Therefore, the evolutionary algorithm for RCPSP implements all GCE principles regarding the population topology, the asynchronous search scheme and the LC, FC, GC societies of individuals. This section focuses on specifying the individual representation, the crossover scheme, the mutation operator and the fitness function used in GCE specifically for RCPSP.

The first thing to consider in an evolutionary algorithm for RCPSP is how a solution of the problem is to be encoded in a chromosome. Several different codifications for RCPSP have been proposed in the literature, among which: activity list representation, random key representation, priority rule representation, shift vector representation and schedule scheme representation. This work focuses on the activity list representation: a solution of the problem is encoded as a list of the activities which represent their execution order. If an activity a_2 appears after another activity a_1 in the activity list, it means that the start time of activity a_2 is higher or equal to the start time of activity a_1: $T(a_1) \leq T(a_2)$. The list of activities must be precedence feasible i.e. each activity must have a higher index than any of its predecessors. An activity list is translated into a schedule by assigning the smallest possible start time to each activity [10]. This means that no activity can be left shifted without violating the constraints.

The Uniform Crossover [4] operator is used for generating offspring in the recombination process. The mutation operator used in the GCE approach to RCPSP swaps two genes of a chromosome provided that the obtained permutation still represents a feasible solution of the problem [4]. Mutation is important in the evolutionary search process as it can lead to new individuals that recombination is not able to obtain.

Finally, the fitness of an individual is defined as the makespan of the corresponding schedule which should be minimized.

5. Computational Experiments and Results

The GCE algorithm is applied to several RCPSP instances and the results are directly compared to a standard evolutionary algorithm (SEA). Both evolutionary algorithms use a population size of 100, mutation rate of 0.05 and 100 generations of evolution. The initial population is randomly generated in such a way that only feasible individuals are obtained. This is done in the following way: (i) the first chromosome gene is randomly selected from the entire activities list, and (ii) next genes are randomly generated from the remaining list of activities as long as all the predecessors of the chosen gene already exist in the chromosome configuration obtained at that point.

In the standard evolutionary approach, roulette selection [4] is used for choosing which individuals should enter the mating pool. The best individual obtained in one generation will always replace one randomly generated individual from the next generation. This mechanism ensures that the best individual obtained in the last generation is actually the best individual obtained in all generations of the algorithm. The GCE model automatically has access to the best individual each generation due to the specific population geometry.

For testing the performance of the proposed model, several ProGen project instances with 60 and 120 activities have been considered [9]. The results obtained after 20 runs of the algorithms for each instance, presented in Table 1 and Table 2, are compaired using the paired t-test with a 95% confidence interval. For the p-values smaller than 0.05 we can conclude that the mean values obtained when using GCE are significantly smaller than those obtained when using SEA. This situation appears for 7 out of the 10 considered project instances with 60 activities and for all 10 considered project instances with 120 activities, even if their complexity is more significant. These test results indicate the acceleration of the search process when using the proposed GCE model.

Table 1. Best and average makespan obtained after 10 runs for each instance with 60 activities

	GCE		SEA		T-test
	Best	Average	Best	Average	p-value
J601_1	81	83.6	85	88	0.007454
J601_2	81	83.6	82	86.9	0.027292
J601_3	73	75.5	75	78.4	0.004937
J601_4	93	95.4	93	98.3	0.045599
J601_5	81	83.5	81	83.2	0.576312
J601_6	66	68.4	71	73.8	0.000067
J601_7	77	80.1	77	82.6	0.136332
J601_8	87	90	89	93.9	0.008539
J601_9	93	97.3	94	97.7	0.583877
J601_10	85	85.8	91	93.4	0.000002

Table 2. Best and average makespan obtained after 10 runs for each instance with 120 activities

	GCE		SEA		T-test
	Best	Average	Best	Average	p-value
J1201_1	140	146.4	154	161.2	0.000296
J1201_2	141	147	150	157.7	0.001782
J1201_3	151	154	158	164.2	0.000560
J1201_4	123	128.7	135	139.1	0.000513
J1201_5	148	153.7	155	166.2	0.000265
J1201_6	105	110.8	116	123.1	0.000011
J1201_7	137	146.5	151	158.4	0.000027
J1201_8	140	150.1	156	160.9	0.000183
J1201_9	133	151.7	160	168.5	0.000718
J1201_10	146	152.2	160	164.7	0.000612

6. Conclusions and Future Work

A geometric collaborative evolutionary model is applied for solving the NP-hard optimization problem of resource-constrained project scheduling. The model is based on the geometric topology and the organization of individuals in different societies with different recombination strategies. The communication between individuals is facilitated by the agent inspired behaviour. Numerical experiments performed on ProGen project instances with 60 and 120 activities emphasize a good performance of the geometric collaborative model. As further work, comparisons with other evolutionary methods for RCPSP will be performed.

7. Acknowledgments

This research is supported by Grant PN II TE 320, Emergence, autoorganization and evolution: New computational models in the study of complex systems, funded by CNCS Romania.

References

[1] J. Blazewicz, J. Lenstra, A. Rinnooy Kan, *Scheduling subject to resource constraints: Classification and complexity.* Discrete Applied Mathematics 5 (1983), 11-24.

[2] C. Chira, A. Gog, D. Dumitrescu, *Exploring Population Geometry and Multi-Agent Systems: A New Approach to Developing Evolutionary Techniques*, GECCO (Companion), 1953-1960, 2008.

[3] C. Chira, A. Gog, D. Dumitrescu, *Asynchronous Collaborative Search using Adaptive Coevolving Subpopulations*, ECOMASS Workshop, Genetic and Evolutionary Computation Conference GECCO'09, F. Rothlauf (Ed), GECCO Companion, ACM, 2575-2582, 2009.

[4] Eiben, A.E., Smith, J.E., *Introduction to Evolutionary Computing*, Springer-Verlag, 2003.

[5] A. Gog, C. Chira, D. Dumitrescu, *Asynchronous Evolutionary Search: Multi-Population Collaboration and Complex Dynamics*, Congress on Evolutionary Computation (CEC 2009), 240-246, 2009.

[6] A. Gog, C. Chira, Recombination operators in permutation-based evolutionary algorithms for the travelling salesman problem, Chapter 10 in Logistics Management and Optimization through Hybrid Artificial Intelligence Systems, IGI Global, (2012) 268-285.

[7] S. Hartmann, *A competitive genetic algorithm for resource-constrained project scheduling.* Tech. Rep. 451, Manuskripte aus den Instituten fur Betriebswirtschaftslehre der Universitat Kiel, (1997).

[8] J.-L. Kim, R.D. Ellis, Jr, *Permutation-Based Elitist Genetic Algorithm for Optimization of Large-Sized Resource-Constrained Project Scheduling*, J. Constr. Eng. Manage. 134, 904 (2008).

[9] R. Kolisch, C. Schwindt, A. Sprecher, *Benchmark instances for project scheduling problems*; Kluwer; Weglarz, J. (Hrsg.): Handbook on recent advances in project scheduling, (1999) 197-212.

[10] R. Kolisch, *Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation*, European Journal of Operational Research, 90, (1996) 320333.

[11] Z. Li, J. Qin, *A new hybrid genetic algorithm and its application in the RCPSP*, Proc. SPIE 8349, 83490Z (2011).

[12] Y. H. Ren, D.C. Kong, W.L. Peng, *A Genetic Algorithm Based Solution with Schedule Mode for RCPSP*, Advanced Materials Research, vol 268-270, (2011), 1802-1805.
[13] H. Zhang, *A Genetic Algorithm for Solving RCPSP*, Computer Science and Computational Technology, 2008. ISCSCT '08. International Symposium on, (2008), 246 249.

Department of Computer Science, Babes-Bolyai University, Kogalniceanu 1, 400084 Cluj-Napoca, Romania
E-mail address: {anca,cchira}@cs.ubbcluj.ro