© International Journal of Combinatorial Optimization Problems and Informatics, Vol. 10, No. 1, Jan-April 2019, pp. 25-31. ISSN: 2007-1558.

Project portfolio selection with scheduling: an evolutionary approach

Daniel A. Martínez-Vega[1], Laura Cruz-Reyes[1], Nelson Rangel V.[2], Claudia G. Gómez S.[1], Patricia Sánchez S.[1], Mercedes Pérez V.[1]
[1]*Tecnológico Nacional de México, Instituto Tecnológico de Ciudad Madero*
[2] *Professor of CONACyT, Division of Postgraduate Studies and Research, Tecnológico Nacional de México, Instituto Tecnológico de Ciudad Madero*

adalbertovega@gmail.com, lauracruzreyes@itcm.edu.mx, cggs71@hotmail.com, jpatricia.sanchez@gmail.com, pvmercedes@gmail.com

Abstract. This paper addresses the project portfolio selection problem enriched with scheduling. The factors considered for project scheduling are: the planning time horizon and the negative impact of the project completion time on the total profit. The objective is to select a subset of projects which maximizes the discounted total gain by late completion time, respecting resource constraints and without exceeding the time horizon. A mixed integer linear programming model was formulated and compared to two recent models addressing the same problem. To show the potential of the model on the large scale we used a metaheuristic based on the genetic algorithm Non-dominated Sorting Genetic Algorithm II. We show experimentally the benefits of our proposal and leave open the possibility of its study applied on a larger scale in future works.

Keywords: Project portfolio selection with scheduling; mixed integer linear programming; genetic algorithm; non-dominated sorting genetic algorithm II.

1 Introduction

In organizations of different turns, there is a common problem, the selection of a project portfolio, whose quality in its solution interferes a lesser or greater degree in their profits, another factor that directly influences these benefits is the completion time of the projects. "In the United States of America, only 26% of information technology projects are carried out on time and within the budget" [1]. As important is the proper selection of projects that integrate the portfolio, as is the efficient timing of such projects. The project portfolio selection problem has been extensively studied, however, the incorporation of temporary dependencies has been little discussed in the specialized literature [2].

This paper addresses the project portfolio selection problem with scheduling. It proposes a linear mathematical model that maximizes the profit and minimizes the time of completion of the projects that integrate the portfolio. We evaluate the quality of our model by contrasting its solutions with those obtained by other models of state of the art. With the purpose of showing the potential of the proposal in large scale, the model was solved with the Non-dominated Sorting Genetic Algorithm II (NSGA-II) metaheuristic.

2 Background

In this section, we describe some basic definitions of the project portfolio selection and scheduling problems, the piecewise linearization method used to improve the solution methodology, and finally a brief description of the NSGA-II algorithm.

2.1 Project portfolio selection problem with scheduling

One of the main tasks of managers in public sector organizations, such as foundations, research centers and companies conducting research and development, is to evaluate a set of projects that compete for financial support, to select those that contribute the maximum benefit to the organization. This subset of projects is a project portfolio [3].

Received Oct 24, 2017 / Accepted Jan 11, 2018

The project portfolio selection problem propounds the following: having a set of N projects, where each $p_i \in N$ has a cost $c_i \in C$ and brings a benefit $b_i \in B$, the question is, is there a subset $X \subseteq N$ such that: $(\sum_{i=1}^{N} x_i c_i) \leq P$, where P is the total budget available?

The complexity of the project portfolio selection problem may increase when considering the management of certain subproblems such as scheduling [2].

The scheduling refers to the organization of time, this feature increases the complexity of the problem and gives a more realistic touch to the basic problem. In the real world, projects require different sizes of lapses to take place, requiring human material, permits and machinery, that in some occasions are requirements that delay the beginning of the fulfillment of a project.

2.2 Piecewise Linearization

Linearization is a procedure that allows approximating a nonlinear model to one that is. Therefore, it meets with the properties of linear systems.

Considering the univariate continuous linear function $f(x)$, where x is within the interval $[a_0, a_m]$, the most used form in linear programming is to approximate the nonlinear function by a piecewise linear function [4].

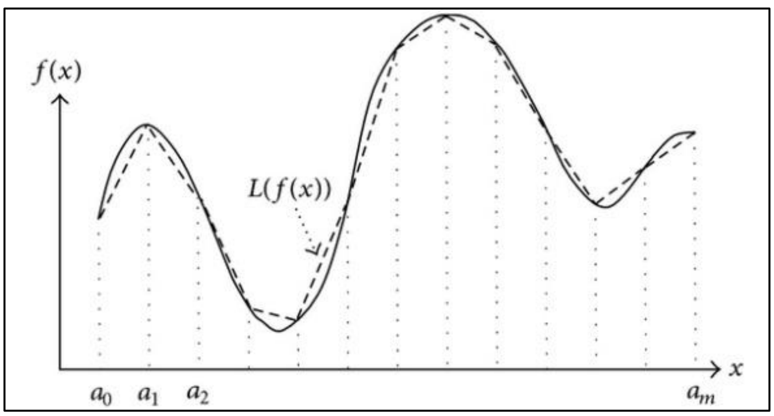

Figure 1. Piecewise Linearization of $f(x)$ [4].

For example, where $a_k (k = 0,1, \dots, m)$ are the break points of $f(x)$, $a_0 < a_1 < \dots < a_m$. Figure 1 shows the graphical representation of this method on $f(x)$, then $f(x)$ can be linearized in the interval $[a_0, a_m]$ as follows:

$$L(f(x)) = \sum_{k=0}^{m} f(a_k) t_k, \quad (1)$$

Where $x = \sum_{k=0}^{m} a_k t_k$, $\sum_{k=0}^{m} t_k = 1$, $t_k \leq 1$, $t_k \geq 0$, and only two adjacent t_k different from zero are allowed. Thus, the linearized function remains as follows [4]:

$$L(f(x)) = \sum_{k=1}^{m} f(a_k) t_k,$$

$$x = \sum_{k=1}^{m} a_k t_k,$$

$$t_0 \leq y_0, \quad (2)$$

$$t_k \leq y_{k-1} + y_k, \text{for } k = 1,2, \dots, m-1, t_m \leq y_{m-1},$$

$$\sum_{k=0}^{m-1} y_k = 1, \sum_{k=0}^{m} t_k = 1,$$

where $y_k \in \{0,1\}, t_k \geq 0, k = 0,1, \dots, m-1$

In the piecewise linearization m binary variables must be added $(y_0, y_1, \dots, y_m)$, the amount of them will be equal to the number of break points $(a_0, a_1, \dots, a_m)$, it should be considered that if m is a very large number it will require a high computational cost. Subsequently, for this case the original function is transformed into a sum of terms in which the new binary variables intervene, the way of intervening will depend on the problem.

2.3 Non-dominated sorting genetic algorithm II

NSGA-II is a metaheuristic which bases its strategy combining the best quality solutions chosen through the grouping of the solution set in *fronts of non-dominance*, employs a method of tournament selection and incorporates solutions that offer diversity; these are selected using the crowding distance method. Algorithm 1 shows all the methods that conform to the NSGA-II metaheuristic, among them the six main ones [5]:

i. *Generation of the population*. It randomly generates the initial population of parents, verifies the feasibility and non-repetition of each member of the population.
ii. *Fast non-dominated sorting*. It forms fronts of non-dominance with the population.
iii. *Selection*. Use the binary tournament strategy to choose a pair of solutions from the parent population. The list of parents adds the solution that belongs to the best front. In case both parents belong to the same front, it selects one randomly.
iv. *Cross-over*. Generates members of the next generation (two for each pair of solutions chosen in the selection method), by SBX cross-over when dealing with real numbers and one-point cross-over when working with binary solutions.
v. *Mutation*. It uses a binary mutation strategy. For the case of real numbers, this algorithm uses a polynomial mutation method.
vi. *Crowding distance*. Method used to provide diversity to solutions by exploring other places in the search space, which otherwise could never be explored, falling into local optimums.

Algorithm 1. NSGA-II [6]

Input: Number of generations n.

Output: P_{t+1}

$Generate(P_t)$
$F = Fast\_non\_dominated\_sorting(P_t)$
$P_t = Tournament\_selection(P_t)$
$Q_t = SBX\_cross\_over(P_t)$
$Q_t = Polinomial\_mutation(Q_t)$
While $(generations < n)$
  $R_t = P_t \cup Q_t$
  $P_{t+1} = \emptyset$ and $i = 1$
  Do
    $P_{t+1} = P_{t+1} \cup F_i$
    $i = i + 1$
  While $|P_{t+1}| + |F_i| \leq N$
  If $(|P_{t+1}| \,!= N)$
    $Crowding\_distance(F_i)$
    $P_{t+1} = P_{t+1} \cup F_i\,[1:(N - |P_{t+1}|)]$
  $F = Fast\_non\_dominated\_sorting(P_{t+1})$
  $Q_{t+1} = SBX\_cross\_over(P_{t+1})$
  $Q_{t+1} = Polinomial\_mutation(Q_{t+1})$
  $t = t + 1$

3 A new mathematical model for project portfolio selection with scheduling problem

The mathematical model proposed in this paper, based on the one proposed by Ghahremani and Naderi in [2], is a linear model unlike the latter. The objective is to maximize the net profit obtained by financing a set of projects (project portfolio), while at the same time finding the order to do them in the shortest possible time that must be less than a pre-established time horizon T.

The project portfolio selection problem when enriching with the scheduling problem [7] changes as follows: to carry out each project it is necessary to develop a set of m activities over the time horizon $T > 0$, which have a predefined order of elaboration, all activities that belong to the same level require the same kind of resource, therefore, only one activity per level can run at the same time.

The proposed mathematical model was generated using a piecewise linearization, in which were added T new binary variables W and T break points indexed by j. The objective and constraint functions obtained from linearization are shown below, the other functions can be consulted in [2]:

The parameters and indices of the model:

- n Number of projects
- m Number of activities
- i Project index, where $i = \{1,2, \dots, n\}$
- j Time index, where $j = \{1,2, \dots, T\}$
- r Discount rate
- T Time horizon
- p_i Benefit of the project i

Decision variables of the model:

- $C_{i,m}$ Continuous variable for the completion time of activity m of project i
- X_i Binary variable that takes the value 1 if project i is selected, and 0 otherwise.
- $W_{i,j}$ Boolean variable for the completion time of project i at time j

Objective function:

$$\max \sum_{i=0}^{n} \sum_{j=1}^{T} p_i W_{i,j} (1+r)^{-j} \tag{3}$$

Subject to:

$$\sum_{j=1}^{T} W_{i,j} = X_i \qquad \forall_i \tag{4}$$

$$\sum_{j=1}^{T} j W_{i,j} = C_{i,m} \qquad \forall_i \tag{5}$$

$$W_{i,j} \in \{0,1\} \qquad \forall_{i,j} \tag{6}$$

Where Equation 4 ensures that only those rows that correspond to the projects included in the portfolio contain only 1 in each. On the other hand, Equation 5 controls that each cell with value 1 matches the value of the column with the completion time of the last activity for the project of the corresponding row. Finally, Equation 6 indicates that the matrix W is of Boolean type because it can only store the values 0 or 1.

It should be noted that this modification is only valid if time is handled as a discrete data.

4 Experimental results

In this section, we present the case of study and the results of the experimentation carried out.

4.1 Experimental evaluation

For the experimentation, we used random instances created using a generator based on the one described in [2]. The quality of the proposed linearized mathematical model was evaluated using its solution with the NSGA-II metaheuristic, and with the ILOG CPLEX integer mathematical programming tool, the results were scaled using the UTC method (Unified Computational Time) [8] for its contrast with those obtained in [1] and [2].

For this experimentation, the solution algorithms were implemented in Java language with NetBeans 8.1, and executed in a machine with the following characteristics: laptop with Intel Core i3 2.1 GHz (2nd generation) processor and 6 GB of RAM. The stop criterion was a maximum execution time of 600 seconds. We used in the experimentation the same parameter configuration as the one used in [2].

4.2 Results

Table 1 presents the comparison of the results obtained when solving the proposed model with NSGA-II, CPLEX, and those presented in [1] and [2].

In addition to presenting the GAP reported by the mathematical tools used, the GAP* is shown. The latter was calculated to have a benchmark that allows comparing by transitivity the results of NSGA-II with those obtained by Ghahremani and Naderi through their experimentation in LINGO and those of Chen in ILOG CPLEX.

As can be seen in Table 1, the performance of our model implemented in CPLEX overcomes those proposed in the literature [1, 2]. Moreover, it can find the optimal solution in instances of larger size in the established time limit of 600s and get closer to the optimal values in a shorter time. On the other hand, the results obtained in NSGA-II are better, besides with a sufficiently wide difference to affirm that, based on our model, it is possible to solve larger scale problems.

Table 1. Experimental results

Size	T	Proposed model in NSGA-II		Proposed model in CPLEX			Ghahremani and Naderi model [1]		Chen model [2]	
		GAP*	Time	GAP*	GAP	Time	GAP	Time	GAP	Time
6 x 3	30	0	0	0	0	0	0	0	0	3
6 x 5	35	0	0	0	0	0	0	6	0	11
8 x 3	35	0	1	0	0	0	0	10	0	30
8 x 5	45	0	2	0	0	0	0	5	14.00%	>600
10 x 3	45	0	15	0	0	226	38.40%	>600	16.22%	>600
10 x 5	60	0.97%	>600	2.85%	6.99%	>600	7.66%	>600	100.00%	>600
12 x 3	80	0.10%	>600	3.34%	9.96%	>600	8.52%	>600	100.00%	>600
Average		**0.15%**		0.88%	2.42%		7.79%		32.88%	

Note: *GAP= optimal value of the objective function – objective function at 600s/optimal value of the objective function*100

Figures 2, 3 and 4 show graphically what is found in Table 1, Figure 2 shows clearly the advantage of the proposed model on the quality of the solutions when it is used within the approximate solver (NSGA-II). In the larger instances, the GAP* is noticeably reduced, which is much smaller than the one obtained in the exact solver generated in CPLEX. It proves the advantage of the approximate solver on the exact solver and allows to deduce its capacity to provide good solutions at instances of larger size, possibly on a large scale.

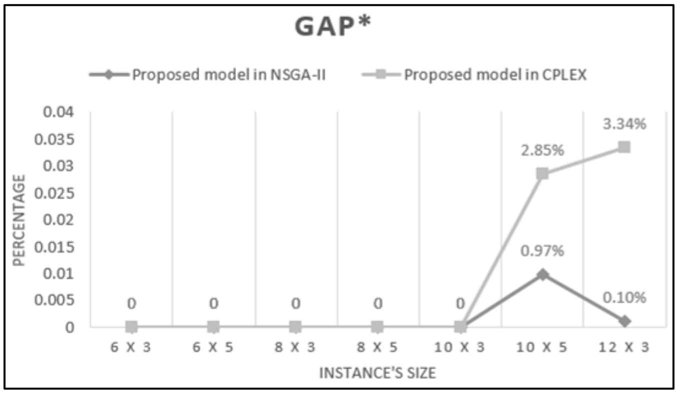

Figure 2. Comparison of the GAP* between our proposed model in NSGA-II and our proposed model in CPLEX.

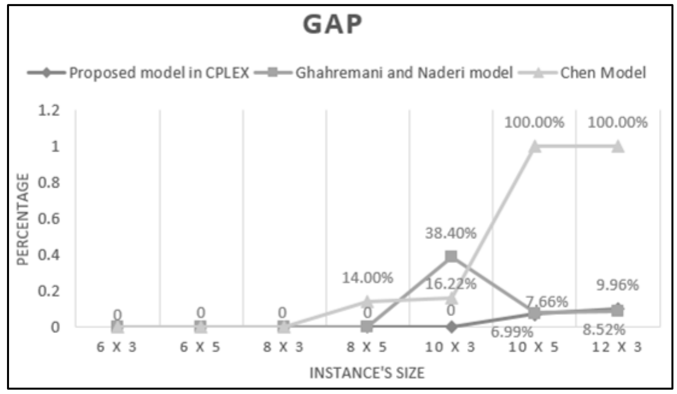

Figure 3. Comparison of the GAP between our proposed model in CPLEX, Gharemani and Naderi model and Chen model.

To obtain an idea of the behavior of the proposed model in the approximate solver on the Ghahremani-Naderi model and the Chen model, in Figure 3 a comparison is made between the GAPs of these three models. Table 1 shows that when using our proposed model in the exact solver, it obtains better results as the size of the instances grows, providing a lower GAP than the other models, indicating that it finds faster solutions nearby to the optimal solution. Finally, a similar comparison to the previous one is shown in Figure 4, with the difference that the execution times are compared instead of the GAPs and it is observed how the model we propose reaches the optimal solution in a greater number of instances within the time limit.

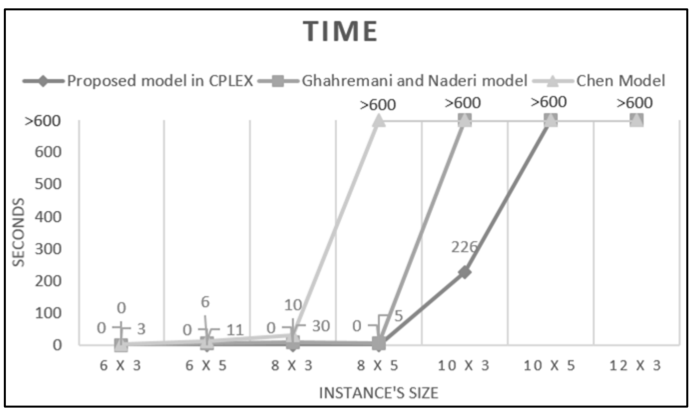

Figure 4. Comparison of the GAP between our proposed model in CPLEX, Gharemani and Naderi model and Chen model.

5 Final comments

In this paper, the study of the project portfolio selection problem with time dependencies was carried out. A model of mixed integer linear programming was presented, focused on the maximization of the total benefits of the problem and the minimization of the execution time of the projects. The proposed model makes the selection of a subset of the projects and schedules them having a time horizon as a restriction. A comparison of our proposed model was made with two models of state of the art. It was found that although linearization increases the number of variables and constraints, this makes our proposal more effective and efficient.

In order to solve problems with a greater number of projects, the solution of our model was implemented based on NSGA-II. The experimental results show that NSGA-II finds optimal solutions in a greater number of instances and in a shorter time than using only a mathematical programming tools such as CPLEX or LINGO.

6 Acknowledgements

This work has been partially supported by: a) Consolidation National Lab Project 280712, Project supported by CONACyT; b) Fronteras de la Ciencias Project 1340, Project supported by CONACyT; c) Project 3058 from the program Cátedras CONACyT; and d) Project 280081 from the program Redes Temáticas Conacyt.

References

1. Chen, J., & Askin, R. G. (2009). Project selection, scheduling and resource allocation with time-dependent returns. *European Journal of Operational Research*, 193(1), 23-34

2. Ghahremani, P., & Naderi, B. (2015). Solution algorithms for the project selection and scheduling problem with resource constraints and time-dependent returns. *International Journal of Industrial and Systems Engineering*, *19*(3), 348-363

3. Nebro, A. J., Luna, F., Alba, E., Dorronsoro, B., Durillo, J. J., & Beham, A. (2008). AbYSS: Adapting scatter search to multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 12(4), 439-457

4. Lin, M. H., Carlsson, J. G., Ge, D., Shi, J., & Tsai, J. F. (2013). A review of piecewise linearization methods. *Mathematical Problems in Engineering*, 2013

5. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, *6*(2), 182-197

6. Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000, September). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *International Conference on Parallel Problem Solving From Nature*(pp. 849-858). Springer Berlin Heidelberg

7. Crainic, T. G., Crisan, G. C., Gendreau, M., Lahrichi, N., & Rei, W. (2009, July). A concurrent evolutionary approach for rich combinatorial optimization. *In Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers* (pp. 2017-2022). ACM

8. Quiroz-Castellanos, M., Cruz-Reyes, L., Torres-Jimenez, J., Gómez, C., Huacuja, H. J. F., & Alvim, A. C. (2015). A grouping genetic algorithm with controlled gene transmission for the bin packing problem. *Computers & Operations Research*, 55, 52-64