Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 817921, 12 pages
http://dx.doi.org/10.1155/2014/817921

Research Article

Comparison of a Fuzzy Genetic and Simulated Annealing Algorithm Approach for Project Time-Cost Tradeoff

Nataša Glišović

Department of Mathematical sciences, State University of Novi Pazar, Vuka Karadžića bb, 36300 Novi Pazar, Serbia

Correspondence should be addressed to Nataša Glišović; natasaglisovic@gmail.com

Received 6 January 2014; Revised 23 June 2014; Accepted 2 July 2014; Published 22 July 2014

Academic Editor: S. H. Nasseri

Copyright © 2014 Nataša Glišović. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Project planning, defining the limitations and resources by leveling the resources available, have a great importance for the management projects. All these activities directly affect the duration and the cost of the project. To get a competitive value on the market, the project must be completed at the optimum time. In other to be competitive enough the optimum or near optimum solutions of time cost tradeoff and the resource leveling and resource constrained scheduling problems should be obtained in the planning phase of the project. One important aspect of the project management is activity crashing, that is, reducing activity time by adding more resources such as workers and overtime. It is important to decide the optimal crash plan to complete the project within the desired time period. The comparison of fuzzy simulated annealing and the genetic algorithm based on the crashing method is introduced in this paper to evaluate project networks and determine the optimum crashing configuration that minimizes the average project cost, caused by being late and crashing costs in the presence of vagueness and uncertainty. The evaluation results based on a real case study indicate that the method can be reliably applied to engineering projects.

1. Introduction

The project has all the elements of business processes and an enterprise that takes place in the future with appropriate risk and uncertainty. The project includes the final objectives to be achieved by involving the limited resources. The project requires the coordination and implementation of techniques while using the network planning. Good functioning of the project as a system depends on how you implement and achieve the objectives defined purposes in a dynamic environment. However, it must be emphasized that the system of this type of work includes the external and internal disturbances. The project is characterized by great complexity, the huge costs, and a large number of participants in the implementation. In order to realize the project effectively, it is necessary to manage it rationally. Because of practical importance, the literature discusses a lot of project management and the tradeoff between duration and total project costs. The special emphasis is placed on the temporal optimization of costs. There are different combinations of choosing the time and the cost of the project activities. The problem is which one is the best. The goal of the optimization time cost is the selection of the best combinations. In the projects, the cost and the duration of activities are variable and uncertain because they depend on many factors (inflation, economic and social conditions, human factors, and natural events). The project and its execution are associated with resources. It is the management of these resources, a better use of the project resources, that can positively affect the profitability of the company.

The project management is the ability to plan and monitor their activities and progress within the cost and time and performance, with the goal of competitiveness. [1] Since the 1950s, the critical path method (CPM) has been demonstrated to be a useful tool in managing the projects in an efficient manner to meet the challenge of managing complicated projects (Kelly, 1961; Siemens, 1971) [2].

In the past, the scheduling of a project (overtime) was done with little planning. The best-known "planning" tool then was the Gantt bar chart, which specifies the starting and finishing times for each activity on the horizontal time scale. Its disadvantage is that the interdependency between the different activities (which mainly controls the progress of the project) cannot be determined from the bar chart. The growing complexities of today's projects have demanded more systematic and more effective planning techniques with the objectives of optimizing the efficiency of executing the project. The efficiency here implies affecting the utmost reduction of the required time to complete the project while accounting for the economic feasibility of using available resources [2].

Shortening the duration of the project may be the result of delays or some other reasons, for the need of an early ending. The project manager, knowing the critical path, with the help of your team can use several techniques to shorten the project. Crashing is a technique for making cost and schedule time tradeoffs to obtain the greatest amount of time compression for the least incremental cost. This technique is also known as time cost tradeoffs.

An important aspect of the project management is the risk management. We are all exposed to daily risk at work and privately and often subconsciously manage risks. This question must be considered actively in the jobs they perform. The modern corporate structure management placed significant emphasis on risk management. The risk can occur in different forms or their activities influence the effectiveness (performance), the various stakeholders, customers/service users, and citizens. It is necessary to understand the way in which these risks can be managed. This is the central part of successful planning and avoidance of failure and achieving goals, building public confidence, and meeting the requirements of good corporate governance. That is the basis of successful planning and avoidance of failure and achieving goals, building public confidence, and meeting the demands of good corporate governance to manage the risks that an organization needs to know what risks it faces and hence evaluate them. Identifying risks is the first step in building a profile of the organization. Identifying risks can be initial and permanent. Once identified, risks should be analyzed and grouped according to a type of threat posed by the objectives of the organization. The risk assessment is based on the collected information about the risk. All actions undertaken in order to solve (reduce) risks can be considered as controls. In every organization there should be an effective screening process and reporting on the risk management in order to perform successful control.

The schedule/time risk essentially implies not completing the project activities on time, resulting in a late completion of the project. The late project completion generally has negative effects on the company such as the penalty costs and customer dissatisfaction. If a project is running late project managers might be able to bring the project back on track by incorporating additional resources. In the project management, this method of mitigating risk is known as crashing.

Sometimes the uncertainty in projects cannot be described by randomness. In fact, the activities of some projects may have been processed many times before, and with historical data, the uncertainty of activity duration time can be described by probability distributions. While the activities of some other projects may be short of statistical data, the duration times can be better described by fuzzy variables. With the development of that search on fuzziness, the fuzzy set theory was also applied into the project scheduling problem, originally by Prade. To the best of our knowledge of the authors, the first work on the fuzzy time cost tradeoff problem was done by Leu et al. In [3], the activity durations were characterized by fuzzy numbers due to environmental variation, and the fuzzy relationship between the activity time and the activity cost was demonstrated by membership function. Furthermore, the philosophy of chance constrained programming was introduced as the decision making criterion. Jin et al. [4] gave a GA-based fully fuzzy optimal time-cost tradeoff model, in which all parameters and variables were characterized by fuzzy numbers and an example in ship building scheduling was demonstrated. Eshtehardian et al. [5] built a multiobjective fuzzy time-cost model, in which fuzzy logic theory was introduced to represent accepted risk levels. Ghazanfari et al. [6] and Ghazanfari et al. [7] applied possibilistic goal programming to the time-cost tradeoff problem to determine optimal duration for each activity in the form of triangular fuzzy numbers. Ke et al. [8, 9] introduced credibility theory into this optimization problem and built three types of fuzzy models. Chen and Tsai [10] constructed membership function of fuzzy minimum total crash cost based on Zadeh's extension principle and transformed the time cost tradeoff problem to a pair of parametric mathematical programs.

Sometimes randomness and fuzziness may exist in the same real project. For instance, Ke and Liu [11] introduced the philosophy of random fuzzy programming into the project scheduling problem. For the complicated environment with randomness and fuzziness, fuzzy random programming is also a very powerful tool for solving time-cost tradeoff problem. Fuzzy random variable was first introduced by Kwakernaak [12, 13].

A fuzzy linear programming (FLP) time-cost tradeoff model is presented and solved by genetic algorithm (GA). The results obtained by GA are compared with those of traditional approach for solving FLP problems. The proposed method solves the time-cost tradeoff problems within uncertain environment and proves suitable through a real project for the Serbian Postal Company.

This paper is organized in four sections including the first section for the introduction of the problem. In Section 2 we represent a project planning problem as a fuzzy optimization problem and present the concept of fuzzy genetic algorithm as a searching tool for obtaining an optimal solution and SA algorithm. In Section 3, we apply the proposed methodology on a real case study. Finally, in Section 4, we conclude this paper with some perspectives on further developments.

2. Basics on the Approach Used for Formulating and Solving the Project Time-Cost Tradeoff Problem

In this section, the main features of the fuzzy approach used in this paper to formulate and solve the project time-cost tradeoff problem are introduced. We divided this section into three subsections: in the first we introduced the main concepts of fuzzy optimization; the second subsection presents the fuzzification model used in this paper; in the third subsection we describe the solution approach and the reasons for using the genetic algorithm.

2.1. Fuzzy Optimization. Fuzzy optimization problems are concerned with the maximization or minimization of single or multiple objectives while satisfying the problem constraints, which represent the model limited resources. Fuzzy optimization's main aim is to find the most satisfying solution (decision alternative) within a fuzzy environment [14].

Bellman and Zadeh addressed the fuzzy optimization problems in their paper on "Decision making in a fuzzy environment" [15] where they consider that in a fuzzy environment there is no distinction between the objectives and the constraints. Following this symmetric fuzzy decision concept, Zimmermann [16] developed the first approach for solving fuzzy linear programming problems. This approach, denoted maxmin, assumes that the decision maker stipulates a goal for each of the objective functions and, hence, both goals and constraints may be considered fuzzy constraints. Further, Zimmermann proposes a crisp equivalent model for maximizing the satisfaction of the intersection of the constraints, which can be solved by traditional methods such as the simplex algorithm. Zimmermann, as Bellman and Zadeh did, points out that using flexible constraints blurs the difference between single and multiple objective problems.

The fuzzification of the single or multiple objective linear programming model usually includes four forms of imprecision [17] (more subtle distinctions are made in [18]:

(i) problems with fuzzy constraints;

(ii) problems with fuzzy objectives (goals imposed on the objective functions);

(iii) pProblems with fuzzy coefficients on the variables;

(iv) combinations of the above.

In this paper we discuss the fuzzification of a linear programming model to deal with project crashing problems in uncertain environments, where we have imprecision in the constraints boundaries and also in the objective function. Therefore, the fuzzy version of this problem with a fuzzy objective and a set of fuzzy constraints, which are of the "less or equal type" or "greater or equal" type, is generally formalized as

$$\begin{gathered}\min \widetilde{Z} = c^T x, \\ Ax \left\{\widetilde{\leq}, \widetilde{\geq}\right\} \widetilde{B}, \\ x \geq 0,\end{gathered} \tag{1}$$

where $\widetilde{Z}$ represents a fuzzy goal, c^T is the vector of crisp costs, A is the matrix that contains the fuzzy coefficients of the objective and of the constraints, and $\widetilde{B}$ is the corresponding vector of the fuzzy limits of the resources.

2.2. Flexible Fuzzification Model. Ribeiro and Pires [19] proposed a method for fuzzy optimization problems that include fuzzy coefficients either in the objective function or in the constraints. The whole process is designated as "flexible approach" for fuzzy optimization problems in its formalization: assuming the formulation of fuzzy optimization problems defined in (1), we made a transformation into the following system of equations:

$$\begin{aligned}
&\min_x \quad Z = c^T x \\
&\max \quad M = \cap\left(\mu_0\left(c^T x\right), \mu_i\left(A_i x\right), \mu_k\left(A_k x\right)\right) \\
&\text{Subject to :} \\
&\qquad c^T x \leq Z_0 + \left(1 - \mu_0\left(c^T x\right)\right) p_0 \\
&\qquad A_i x \leq b_i + \left(1 - \mu_i\left(A_i x\right)\right) p_i, \quad (i = 1, \ldots, m) \\
&\qquad A_k x \geq b_k + \left(1 - \mu_k\left(A_k x\right)\right) p_k, \quad (k = 1, \ldots, n) \\
&\qquad \left(\mu_0\left(c^T x\right), \mu_i\left(A_i x\right), \mu_k\left(A_k x\right)\right) \in [0, 1] \\
&\qquad x \geq 0.
\end{aligned} \tag{2}$$

It should be noted that all the fuzzy parameters are presented with a tilde on the top. The ∩ symbol represents the intersection of all membership values. The second objective represents the maximization of the intersection of all membership value; that is, this goal gives the best value of the minimum (intersection) of the violations constraints.

The most frequently used membership functions to represent the deviations accepted for the fuzzy objectives (goals), constraints, and/or coefficients are triangular functions. In particular, let μ_0 denote the membership functions for the ith or kth constraint. Let p_0 and p_i and p_k be the permissible tolerances for the objective function and the constraints. Then we may decide μ_0 and μ_i or μ_k to be nonincreasing or nondecreasing as well as continuous membership functions as follows:

$$\mu_i = \begin{cases} 1 & A_i x \leq b_i \\ 1 - \dfrac{A_i x - b_i}{p_i} & b_i < A_i x \leq b_i + p_i \\ 0 & A_i x > b_i - p_i, \end{cases} \tag{3}$$

$$\mu_k = \begin{cases} 1 & A_k x \geq b_k \\ 1 + \dfrac{A_k x - b_k}{p_k} & b_k - p_k \leq A_k x < b_k \\ 0 & A_k x < b_i - p_i, \end{cases} \tag{4}$$

$$\mu_0 = \begin{cases} 1 & c^T x < Z_0 \\ 1 + \dfrac{Z_0 - c^T x}{p_0} & Z_0 \leq c^T x \leq Z_0 + p_0 \\ 0 & c^T x > Z_0 + p_0. \end{cases} \tag{5}$$

Function (3) corresponds to the membership function for constraints "less or equal," function (4) is the membership function for the constraints of the “greater or equal” type, and function (5) is the membership function for objective function of minimization type.

2.3. The Concept of Genetic Algorithm.

The GAs, developed by Holland in the 1960s and 1970s, are based on the fundamental concept of Darwin-type natural evolution. Gene selection is a process that is based on the principles of adequacy imposed by individual outer world (not the strongest or the fastest one that survives). Implicitly, the idea is the kind of progressive improvement over previous generations. Between two successive generations, there are small changes, but they are favorable to accumulate over many generations; they produce big changes. GA operates on the principle of accumulation of generations until the convergence level that is considered optimal. By evaluation it selects the best individuals of the population and by their reproduction, recombination of their genes, you get the population. The objective of GAs is to generate successive populations of solutions whose roots (genes) are selected from the most promising (highest accurate) solutions of the previous population. GAs combine random search procedures with highly effective exploration techniques.

The genetic algorithm is a family of algorithms that use some of the genetic principles that exist in nature, and all for the purpose of the solutions of certain computational problems. Those natural principles are inheritance, crossing, mutation, survival of the fittest, migration, and so on. These algorithms are used for solving various classes of problems, especially in the optimization-finding optimal parameters of a system. The genetic algorithm searches stochastically and therefore is used with the search of the extreme of space of large dimensions. Applying the mechanisms of genetic algorithms, the information of these local extremes is exchanged, and the aim of the algorithms is not to be finished in one of them, but to recognize among them the one which is also global at the same time. Exactly these qualities of the genetic algorithm are the reason why this method is used in the paper. That is why we need a model which effectively leads to conclusions and helps in making better decisions for the company, all for the purpose of better business and therefore the ranking of companies.

The fundamental operators of GAs are as follows:

(a) the selection and reproduction of better individuals;

(b) genetic recombination (crossovers);

(c) the random mutation of individuals genes.

The selection is a fundamental operation which is performed with the aid of a function for the evaluation of the fitness of individuals. The straightforward duplication of selected individuals, reproducing them identical to those of the previous population, entails no benefit in terms of the exploration of solution spaces.

The recombination of genes can be accompanied by the operation of mutation, which has a very low probability, so as not to destroy entirely the genetic heritage accumulated through previous selections. Performed in this way, the mutation makes it possible to enrich the variety of individuals present in the population, preventing them from tending to be too uniform, hence losing the capacity to evolve [20].

The fitness function is designed to evaluate the performance of the individuals (e.g., functions) who make up the populations, transforming the fitness of the solutions proposed by the GAs into numeric values based on their performance. The typical steps into which a GA procedure proceeds are as follows:

(a) the initial population of individuals (or genomes) is generated randomly (by random numbers, binary strings of zeroes add ones are generated, all with the same length);

(b) for every individual, fitness is calculated in relation to the problem that has to be solved (i.e., the good hypothetical solutions);

(c) the degree of homogeneity of the entire populations fitness is calculated (bias);

(d) individuals are ordered on the basis of their fitness and those suitable to generate the subsequent population are selected;

(e) the successive population is generated on the basis of reproduction of new individuals, starting from the ones selected in the previous population;

(f) in the new population, the sequence is repeated from point (b) onwards.

The evolutionary strategy has the theoretical ability of finding the global optimum, but the success of the approach strongly depends on some parameters. These parameters have to be predetermined and influence the result of the optimization as well as the computation time needed. The most important parameters to be chosen are as follows:

(i) number of parents selected (μ): the more robust the algorithm gets, the more parents are selected, but also needs more generations to find the solution, because strong improvements are diluted by recombination;

(ii) number of descendants created (λ): the number of descendants influences the ability of the algorithm to find the optimal solution; the more descendants are created, the more robust the algorithm gets, but also the number of function evaluations increases;

(iii) standard deviation (σ): the standard deviation of the normal distribution influences the search region; for large standard deviations the algorithm has the ability to search a larger region; this enables algorithm to escape local optima; Near the optimal solution only small steps lead to a further improvement;

(iv) stopping criterion (ε): as no gradient information is available as a convergence criterion, a stopping criterion has to be formulated; some possibilities are maximum number of iterations or the best individual stays the same for a given number of iterations or the difference between the best and the worst individual of one generation is below a certain stopping criterion.

```
Algorithm for solving fuzzy optimization model
  /* **Initialization** */
  set $N_c$ = number constraints
  select an initial state $x$
  /* **Membership determination** */
  **for** $i = 1$ **to** $N_c$
        $\mu(i)$:= membership value of the constraint $R_i(x)$
  min1:= aggregation($\mu(1), \ldots, \mu(N_c)$)
  /* **Genetic Algorithm** */
  **set** numberOfGeneration = 0
  **repeat**
        **repeat**
              createNextGeneration
              **for** $i = 1$ **to** $N_c$
                    $\mu$(i):= membership value of the constraint $R_i(x)$
              min2 = aggregation($\mu(1), \ldots, \mu(N_c)$)
              calculate delta = min2 − min1
              if delta > 0 then
                    UpdateNextPopulation;
              min2 = min1;
        **until** not terminate($y$, evaluate($y$), $\theta(k)$)
     numberOfGeneration++;
  **until** stop criteria
```

Algorithm 1: Genetic algorithm pseudocode for maxmin approach.

The membership functions are then built with the tolerances defined. Furthermore, the GA algorithm allows dealing with any type of fuzzification.

When we use the GA algorithm for solving fuzzy optimization problems, the decision maker can select the thresholds levels (α-cuts) as well as the tolerances (deviations) for each constraint parameter and/or for each objective function's coefficient and/or for each constraint's coefficient. The membership functions are then built with the tolerances defined. Furthermore, the GA algorithm allows dealing with any type of fuzzification.

The objective of the genetic algorithm implementation in this paper is to solve fuzzy optimization model presented in the previous chapter by maximizing the aggregation of the membership values of the goals and constraints. Its implementation is defined by following (Algorithm 1).

The procedure *createNextGeneration* uses the four external functions with the intuitive meaning:

let x' = recombine (x, theta(r));

let x'' = mutate (x', theta(m));

let pom = evaluate(x'');

y = select (x'', pom, μ, theta(s)).

At the recombine function, at the beginning, $N_{\text{pop}}/2$ pairs f individuals are chosen (N_{pop} is a population number). Then, for each pair of individuals, with the given probability, in advance, of theta (x) recombination, in an accidental way, a position is chosen for the recombination and the exchange of bit genetic codes of the given pairs of individuals, after the given position, has been carried out. Schematically the following can be shown.

Before the recombination:

XXX|XXXXX
YYY|YYYYY

After recombination:

XXX|YYYYY
YYY|XXXXX

The mutation function for the parameter input has a level of mutation (theta(m)), that is, the probability and the individual x', while by the accidental choice the mask is generated and, as a result, it returns the mutated individual x''. For example consider the following.

If the level of the mutation theta(m) = 0.2 and individual x' = 010111000101 and if by the accidental choice the generated mask is 001000000100, then

the mask is

001000000100

the individual before the mutation (x') is

010111000101

the individual after the mutation (x'') is

011111000001

```
/* Algorithm for solving SA-fuzzy optimization model */
set $N_c$ = number constraints
set count temperature $t = 0$
set initial temperature $T > 0$
set the threshold $\alpha > 0$
select an initial state $x$
for $i = 1$ to $N_c$
    $\mu(i)$:= membership value of the constraint and coefficient of objective function $R_i(x)$
    min1:= aggregation($\mu(1), \ldots, \mu(N_c)$)
repeat
    for $i = 1$ to $N_c$
        $\mu$(i):= membership value of the constraint and coefficient of objective function $R_i(y)$
    min2:= aggregation($\mu(1), \ldots, \mu(N_c)$)
    calculate $\Delta$ = min2 − min1
    if $\Delta > 0$ then
        $X = Y$;
        min1 = min2;
    else
        Create random number $p$ = rand(0, 1)
        if exp($\Delta/T$) > $p$
            $X = Y$;
            min1 = min2;
        else
            $Y = X$;
            min2 = min1;
    $t = t + 1$
until $t > T$
```

Algorithm 2

At the selection function, a simple roulette selection has been used, and in an accidental way (theta(s)), "good" solutions (y) are received from the population.

The procedure *updateBestPopulation* is used for providing the best population found at the moment:

$$\text{if (evaluate}(y) < \text{evaluate}(x)\text{), then } x = y.$$

2.4. The Concept of Simulated Annealing. The SA is a search technique that belongs to the general class of guided random search techniques [21] which are based on enumerative techniques but use additional information to guide the search. Guided random search techniques are quite general in scope and can solve complex problems. This type of algorithms play an important role in the scope of local search techniques, for two major reasons [22]: first, because they have been proved successful when applied to a large number of practical problems; second, some of those algorithms, such as the SA algorithms, have a stochastic component, which facilitates a theoretical analysis of its asymptotic convergence and this made them quite popular among mathematicians. The objective of an algorithm of this nature is to find the best solution among a finite number of possible solutions. The SA technique is particularly attractive, because it allows finding near-optimal solutions with reasonable computational effort. Notice that in this algorithm it is not possible to know if the best solution found is indeed the global optimum. This characteristic restricts its use to the cases where good local optima are acceptable [23, 24]. The SA algorithm is an algorithm used for the solution of optimization problems, where the objective function corresponds to the energy of the states of a solid. The SA algorithm requires the definition of a neighborhood structure, as well as the parameters for the cooling programming. A temperature parameter allows the distinguishing among deep or slight alterations in the objective function. Drastic alterations occur at high temperatures and small or slight modifications at low temperatures. It is an "evolutionary" process that moves in small steps, from a state to another, and there exists the problem of being "arrested" by a local optimum. When we use the SA algorithm for solving fuzzy optimization problems, the decision maker can select the thresholds levels (α-cuts) as well as the tolerances (deviations) for each constraint parameter and/or for each objective function's coefficient and/or for each constraint's coefficient. The membership functions are then built with the tolerances defined. Furthermore, the SA algorithm allows the dealing with any type of fuzzification. The objective of the simulated annealing algorithm implementation in this paper is to solve fuzzy optimization model presented in the previous chapter by maximizing the aggregation of the membership values of the goals and constraints.

Its implementation is defined by Algorithm 2.

Min represents the aggregation of the memberships of the fuzzy constraints. As aggregation operator, the t-norm min is used. The variable Δ represents the difference between the previous Min and new Min. This represents the old and new energy states difference. The probability function of moving to a smaller energy state is given by exponential of Δ divided by the control parameter T. The smaller the temperature, the less probability any change will occur. The algorithm includes also $N(t)$ as the number of generated neighbors. $T(t)$ is decreasing function of temperature. The algorithm stops when the temperature is less than a defined limit value. As presented in previous chapter, the membership functions of the fuzzy goal and constraints used in this application are triangular functions [25].

The main disadvantage of the SA algorithm is the need to define the initial states that satisfy the constraints, for each variable. Another disadvantage is the choice of an appropriate initial temperature value (T), because it can imply a longer search and consequently more computation time. Relatively to the evaluation factor, time, it is convenient to refer that the solution time of the algorithm did not show and clearly predictable behaviour, but a behaviour that depended, strongly, on the problem. This nature of convergence is a typical aspect of nondeterministic search methods, such as the SA [23].

The decreasing function of temperature in this research was chosen based on testing of the implemented model through the program in C#. For these selected parameters a better optimization is obtained.

3. Fuzzy Genetic and SA as a Tool for Project Time-Cost Tradeoff Problem Solving

Let us now introduce a linear programming formulation for the optimal project crashing problem. Consider a project network $S = (V, P, t, c)$ consisting of a finite set V of nodes (events) and a set $P \subset V \times V$ of arcs with crisp activity times, which are determined by functions $t : P \to R^{+}$ and $c : P \to R^{+}$, and attached to the arcs. Parameter t_{ij} represents the duration of an activity $(i, j) \in P$ and c_{ij} is the cost per unit time of reducing the duration of the activity $(i, j) \in P$. The purpose of the time-cost tradeoff analysis of project is to determine which activities in the network to shorten and by how much such that the total crash cost is minimized while achieving the desired project completion time at the same time [10].

In the time-cost tradeoff analysis of project networks two types of costs associated with each activity $(i, j) \in P$ are estimated; normal costC^{n}_{ij} and crash costC^{c}_{ij}, associated with normal time T^{n}_{ij} and crash time, estimate T^{c}_{ij} for each activity $(i, j) \in P$. The relationship between the normal cost and the crash cost with respect to activity time is in this paper assumed to be linear:

$$\Delta C_{ij} = \frac{C^{c}_{ij} - C^{n}_{ij}}{T^{n}_{ij} - T^{c}_{ij}}. \tag{6}$$

Let the duration of activity time T_{ij} be determined to be shortened for ΔT_{ij}. By employing the relationship of (6), the incremental direct cost for each activity $(i, j) \in P$ is $\Delta C_{ij} \Delta T_{ij}$; thus, the total project cost of crashing activities is $\sum_{i=1}^{n} \sum_{i=1}^{n} \Delta C_{ij} \Delta T_{ij}$.

Therefore, the objective of the time-cost tradeoff analysis is to minimize the total crash cost that can be expressed as

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} \Delta C_{ij} \Delta T_{ij}. \tag{7}$$

Consequently, the time-cost tradeoff problem of a project network with n nodes can be formulated as the following LP:

$$Z = \min \sum_{i=1}^{n} \sum_{j=1}^{n} \Delta C_{ij} \Delta T_{ij}, \tag{8}$$

subject to

$$\Delta T_{ij} \le T^{n}_{ij} - T^{n}_{ij}, \quad i = 1, \dots, n, \quad j = 1, \dots, n, \; i \ne j, \tag{9}$$

$$T_{j} - \Delta T_{ij} - T_{i} \ge T_{ij}, \quad i = 1, \dots, n, \quad j = 2, \dots, n, \; i \ne j, \tag{10}$$

$$T_{n} \le \mathrm{PCT}, \tag{11}$$

where T_{i} is the earliest time for node i and PCT is the indicated project completion time. In this model, constraints (9) represent crash time constraints. The time to complete an activity can be reduced by increasing the resources or by improving the productivity which also require additional resources. But it is not possible to reduce the time required to complete an activity after a certain threshold limit. Constraints (10) describe the network. As we mentioned, activities of a project are interrelated; the starting of some activities is dependent upon the completion of some other activities. Since event 1 will start at the beginning of the project, we begin by setting the occurrence time for event 1 to equal to zero. Constraint (11) is the project completion constraint. This constraint will recognize that the last event must take place before the project deadline date. However, in real world for many projects we have to use human judgment for estimating. A way to deal with this imprecise data is to employ the concept of fuzziness.

3.1. Fuzzy GA and SA Approach Applied to a Real Case Study.

To illustrate the validity of the proposed approach, the project "Main Postal Centers on the territory of Republic of Serbia" is investigated. Table 1 contains all the activities and corresponding time and cost parameters. To model the problem, we need to define all the activities in terms of starting and ending points (Figure 1). The project network has 18 activities and 15 events.

According to the critical path method (CPM), for normal length of activities, there are two critical paths, A-C-E-I-M-P-R and A-C-G-K-M-P-R, and the time length on these paths is 1180 time units (days). For the case of crash time of activities, the situation concerning the critical paths remains the same, while the time length is now 1010 days.

Assume that we want to adjust the deadline of the project on its expected crash completion date. According to the

Table 1: Activities, time, costs, and relationships for the considered project network.

Activity	Description	Depends on	Normal time (days)	Crash time (days)	Normal cost (EUR)	Crash cost (EUR)
A	Conceptual design for the complex of Main Postal Center (MPC) of Belgrade	—	90	70	3700	5000
B	Technical-technological concept of MPC Belgrade complex	—	30	25	2900	3200
C	Providing all necessary terms and agreements for communal connections	A	120	100	5000	6500
D	Conducting an open public acquisition procedure for the service of technological documentation produced for MPC Belgrade construction	B	60	60	1000	1000
E	Making the investment-technical documentation for construction of MPC Belgrade	C, D	120	110	3522300	3700000
F	Making the investment-technical documentation for construction of MPC Nis	C, D	60	55	23600	29200
G	Making the investment-technical documentation for construction of MPC Novi Sad	C, D	120	100	153400	160000
H	Making the technological conception	C, D	90	80	362850	387300
I	Construction of MPC Belgrade	E	360	320	15616120	15970000
J	Reconstruction of MPC Nis	F	120	110	1077340	1090750
K	Reconstruction and upgrading of MPC Novi Sad	G	360	320	1437240	1500000
L	Equip the objects: office equipment	I, J, K, H	90	80	11800	15000
M	Equip the objects: technological equipment	I, J, K, H	100	80	15300	18000
N	Equip the objects: IT equipment	I, J, K, H	80	60	5900	7000
O	Installing the equipment for automatic sorting of letter posts	L, M, N	120	110	7860000	8000000
P	Installing the equipment for automatic sorting of printed matters (flats)	L, M, N	270	240	8950000	9160000
Q	Installing the equipment for automatic sorting of packets	L, M, N	120	100	6774700	6944000
R	Installing the IT and telecommunication equipment	O, P, Q	120	90	11800	13200

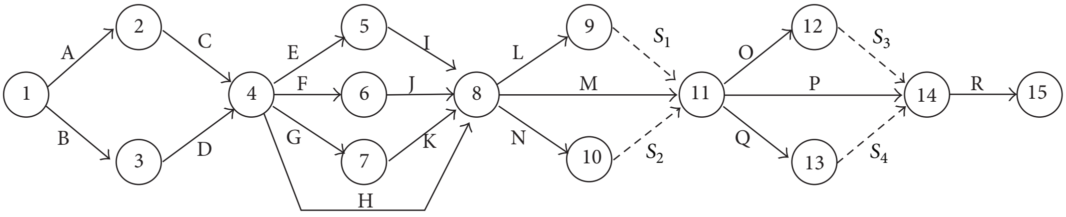

Figure 1: Project network diagram.

LP formulations (8)–(11), the optimal solution of time-cost tradeoff problem for our project is given in Table 3.

However, in reality there are a number of unpredictable situations that may affect the activity times of a project. Therefore, to reflect the real situations during the project execution we should relax the crashing time and network constraints. Hence, we will now fuzzify the problem, considering the tolerance intervals for the objective function and all constraints (Table 2). In this table, the tolerance intervals have been given for everyone of the constraints (constraints of "less or equal" type and constraints of "greater or equal" type). For every of the costs c_i, the tolerance d_i has been given and on

Table 2: Permissible tolerances for project time-cost tradeoff problem.

Constraints of “less or equal” type			Constraints of “greater or equal” type		
Lower bound (c_i)	Deviation (d_i)	Upper bound ($c_i + d_i$)	Lower bound ($c_i - d_i$)	Deviation (d_i)	Upper bound (c_i)
$Z = c_0 = 750$	220	970	85	$d_{25} = 5$	$c_{25} = 90$
$c_1 = 20$	$d_1 = 2$	22	20	$d_{26} = 10$	$c_{26} = 30$
$c_2 = 5$	$d_2 = 3$	8	110	$d_{27} = 10$	$c_{27} = 120$
$c_3 = 20$	$d_3 = 3$	23	40	$d_{28} = 20$	$c_{28} = 60$
$c_4 = 0$	$d_4 = 5$	5	115	$d_{29} = 5$	$c_{29} = 120$
$c_5 = 10$	$d_5 = 3$	13	55	$d_{30} = 5$	$c_{30} = 60$
$c_6 = 5$	$d_6 = 5$	10	118	$d_{31} = 2$	$c_{31} = 120$
$c_7 = 20$	$d_7 = 5$	25	350	$d_{32} = 10$	$c_{32} = 360$
$c_8 = 10$	$d_8 = 5$	15	115	$d_{33} = 5$	$c_{33} = 120$
$c_9 = 40$	$d_9 = 5$	45	352	$d_{34} = 8$	$c_{34} = 360$
$c_{10} = 10$	$d_{10} = 5$	15	85	$d_{35} = 5$	$c_{35} = 90$
$c_{11} = 40$	$d_{11} = 5$	45	85	$d_{36} = 5$	$c_{36} = 90$
$c_{12} = 10$	$d_{12} = 5$	15	77	$d_{37} = 3$	$c_{37} = 80$
$c_{13} = 10$	$d_{13} = 5$	15	98	$d_{38} = 2$	$c_{38} = 100$
$c_{14} = 10$	$d_{14} = 5$	15	0	$d_{39} = 0$	$c_{39} = 0$
$c_{15} = 10$	$d_{15} = 5$	15	0	$d_{40} = 0$	$c_{40} = 0$
$c_{16} = 40$	$d_{16} = 5$	45	115	$d_{41} = 5$	$c_{41} = 120$
$c_{17} = 20$	$d_{17} = 5$	25	117	$d_{42} = 3$	$c_{42} = 120$
$c_{18} = 30$	$d_{18} = 5$	35	0	$d_{43} = 0$	$c_{43} = 0$
$c_{19} = 30$	$d_{19} = 7$	37	0	$d_{44} = 0$	$c_{44} = 0$
$c_{20} = 0$	$d_{20} = 5$	5	117	$d_{45} = 3$	$c_{45} = 120$
$c_{21} = 0$	$d_{21} = 5$	5			
$c_{22} = 0$	$d_{22} = 5$	5			
$c_{23} = 0$	$d_{23} = 5$	5			
$c_{24} = 1010$	$d_{24} = 40$	1050			

the basis of it, the tolerance intervals have been determined (upper bound and lower bound).

We solved this fuzzy linear programming problem using Zimmermann's approach, which considers that in a fuzzy environment there is no difference between goals and constraints. Within this approach the objective moves into a constraint with a defined stretched boundary as any other fuzzy constraint. The resolution process proceeds to transform the problem into a crisp one, using the objective for the maximization of the minimization of the membership values of the constraint deviations, determined in functions like those in expressions (3)–(5). The results of the problem by using fuzzy Zimmermann approach are given in Table 3.

Now, on the base of flexible fuzzification model presented in Chapter 2, we developed a suitable genetic algorithm for solving the model for considered project crashing problem. For the implementation of the GA algorithm we have to specify the following: (a) how to generate a state y (create NextGeneration), (b) which aggregation function to use, (c) the selection of number of neighbours to generate (update BestPopulation), and (d) the algorithm stopping criteria.

The stopping criterion used is reached when number of generation is 500. Fuzzy GA algorithm for project time-cost tradeoff solving is coded in C programming language.

We tested our approach for different threshold values. Obtained results are compared with the crisp version (A), Zimmermann's approach (B), and the GA algorithm based on flexible fuzzification model (C) for the considered project time-cost tradeoff problem.

Using the crisp transformation of Zimmermann to relax the objective function and the constraints, we obtained the value of 754.03 EUR, which represents a decrease for 179.97 EUR as costs of project time crashing comparing to the crisp case.

Applying our fuzzy GA approach for the threshold value $\alpha = 0.94$ for all constraints we obtained slightly worse results for the objective function in comparison with Zimmermann's approach, but still well below the defined goal limit. Comparing decision variable values with those from fuzzy LP solution, there are differences in most crashing time constraints, but all these new values are within the boundaries, so there is no infeasibility.

The threshold $\alpha = 0.98$ provides the best results, as we can see from Table 3. Applying $\alpha = 0.98$ threshold we

Table 3: Solutions of the Zimmermann's approach and flexible fuzzification model for threshold $\alpha = 0.98$.

Variable	A	B	C	Variable	A	B	C	Variable	A	B	C
$Z=$	934	760.25	754.03								
M		0.95	0.98								
Δt_{12}	0.00	0.00	0.00	Δt_{1112}	0.00	0.00	0.02	t_7	330	328.97	328.97
Δt_{13}	0.00	0.00	0.00	Δt_{1114}	0.00	0.00	0.01	t_8	690	688.60	688.30
Δt_{24}	0.00	0.00	0.10	Δt_{1113}	0.00	0.00	0.04	t_9	790	788.74	788.74
Δt_{34}	0.00	0.23	0.23	Δt_{1415}	20.00	16.13	1.76	t_{10}	790	768.40	769.40
Δt_{45}	0.00	0.00	0.00	Δt_{911}	0.00	0.23	0.24	t_{11}	790	788.51	788.51
Δt_{46}	0.00	0.00	0.00	Δt_{1011}	0.00	0.23	0.40	t_{12}	910	908.37	908.37
Δt_{47}	0.00	0.23	0.03	Δt_{1214}	0.00	0.23	0.10	t_{13}	910	908.37	908.37
Δt_{48}	0.00	0.00	0.00	Δt_{1314}	0.00	0.23	0.50	t_{14}	910	908.14	918.14
Δt_{58}	0.00	0.00	0.00	t_1	0.00	0.00	0.00	t_{15}	1010	1011.87	1041.87
Δt_{68}	0.00	0.00	0.01	t_2	90.00	89.77	90.77				
Δt_{78}	0.00	0.00	0.00	t_3	150.00	29.53	29.53				
Δt_{89}	0.00	0.00	0.00	t_4	210.00	209.30	209.30				
Δt_{811}	0.00	0.00	0.02	t_5	330.00	329.07	329.07				
Δt_{810}	0.00	0.00	0.02	t_6	570.00	568.83	568.83				

obtained a value 754.03 for the objective function, which is improvement in relation to fuzzy Zimmermann (see Table 3). Table 3 shows the obtained results which are compared with the crisp version (A), Zimmermann's approach (B), and the GA algorithm based on flexible fuzzification model (C) for the considered project time-cost tradeoff problem. Using the crisp transformation of Zimmermann to relax the objective function and the constraints, we obtained the value of 754.03 EUR, which represents a decrease for 179.97 EUR as costs of project time crashing comparing to the crisp case. Comparing decision variable values we can observe that values for the variable t_8 present a reduction in time, relatively to both of alternative approaches, while the t_2, t_{15} have slightly higher values.

In summary we can conclude that the use of genetic algorithm for the purpose of solving the project time-cost tradeoff problem in the presence of uncertainty seems quite more flexible and adaptable than the crisp mathematical transformation defined in Zimmermann method.

As we can see from Table 4, the fuzzy GA, as a tool for project crashing problem provides not only the aggregated value but also all the constraint violations. In the case if it is used in a situation where thresholds can be defined, the decision maker can select the degree of violation allowed for each constraint.

This heuristic (SA) takes under consideration that more successors should be generated when the temperature decreases in order to have more options to test. The temperature function $T(t)$ is a function that uses a decreasing factor of 0.95. The stopping criterion used is reached when the temperature is less than 0.002. Fuzzy SA algorithm for project time-cost tradeoff solving is coded in C programming language.

We tested our approach for different threshold values. Obtained results are compared with the crisp version (A), Zimmermann's approach (B), and the SA algorithm based on flexible fuzzification model (C) for the considered project time-cost tradeoff problem.

Table 4: Min results.

Variables	GA approach	
	Maxmin ($\alpha > 0.94$)	Maxmin ($\alpha > 0.98$)
Z	760.25	754.03
$Ax \,\widetilde{\geqq}\, B$	0.941	0.985
$Ax \,\widetilde{\leqq}\, B$	0.941	0.985
Min	0.941	0.985

Using the crisp transformation of Zimmermann to relax the objective function and the constraints, we obtained the value of 755.31 EUR, which represents a decrease for 178.69 EUR as costs of project time crashing comparing to the crisp case.

Applying our fuzzy SA approach for the threshold value $\alpha = 0.94$ for all constraints we obtained slightly worse results for the objective function in comparison with Zimmermann's approach, but still well below the defined goal limit. Comparing decision variable values with those from fuzzy LP solution, there are differences in most crashing time constraints, but all these new values are within the boundaries, so there is no infeasibility.

The threshold $\alpha = 0.98$ provides the best results, as we can see from Table 5. Applying $\alpha = 0.98$ threshold we obtained a value 755.31 for the objective function which is improvement with respect to fuzzy Zimmermann. Comparing decision variable values we can observe that values for variables t_9, t_{15} present a reduction in time, relatively to both of alternative

Table 5: Solutions of the Zimmermann's approach and fuzzy SA model for threshold $\alpha = 0.98$.

Variable	A	B	C	Variable	A	B	C	Variable	A	B	C
$Z =$	934	762.89	755.31								
M		0.95	0.98								
Δt_{12}	0.00	0.00	0.00	Δt_{1112}	0.00	0.01	0.02	t_7	330	328.97	328.97
Δt_{13}	0.00	0.00	0.00	Δt_{1114}	0.00	0.00	0.01	t_8	690	688.30	688.30
Δt_{24}	0.00	0.10	0.10	Δt_{1113}	0.00	0.00	0.04	t_9	790	788.74	784.74
Δt_{34}	0.00	0.23	0.23	Δt_{1415}	20.00	13.86	1.76	t_{10}	790	768.40	768.40
Δt_{45}	0.00	0.00	0.00	Δt_{911}	0.00	0.24	0.24	t_{11}	790	788.51	788.51
Δt_{46}	0.00	0.00	0.00	Δt_{1011}	0.00	0.40	0.40	t_{12}	910	908.37	908.37
Δt_{47}	0.00	0.03	0.03	Δt_{1214}	0.00	0.10	0.10	t_{13}	910	908.37	908.37
Δt_{48}	0.00	0.01	0.01	Δt_{1314}	0.00	0.50	0.50	t_{14}	910	918.14	918.14
Δt_{58}	0.00	0.00	0.00	t_1	0.00	0.00	0.00	t_{15}	1010	1021.87	1011.87
Δt_{68}	0.00	0.01	0.02	t_2	90.00	89.77	90.77				
Δt_{78}	0.00	0.00	0.00	t_3	150.00	29.53	29.54				
Δt_{89}	0.00	0.00	0.00	t_4	210.00	209.30	209.30				
Δt_{811}	0.00	0.01	0.01	t_5	330.00	329.07	329.07				
Δt_{810}	0.00	0.02	0.03	t_6	570.00	568.83	568.83				

Table 6: Min results.

	Simulated annealing approach	
Variables	Maxmin ($\alpha > 0.94$)	Maxmin ($\alpha > 0.98$)
Z	762.89	755.31
$Ax \ \tilde{\geqq}\ B$	0.941	0.985
$Ax \ \tilde{\leqq}\ B$	0.941	0.985
Min	0.941	0.985

approaches, while the t_2, t_3 have slightly higher values (see Table 5).

In summary we can conclude that the use of Simulated Annealing for the purpose of solving the project time-cost tradeoff problem in the presence of uncertainty seems quite flexible and adaptable than the crisp mathematical transformation defined in Zimmermann method.

As we can see from Table 6, the fuzzy SA, as a tool for project crashing problem provides not only the aggregated value but also all the constraint violations. In the case if it is used in a situation where thresholds can be defined, the decision maker can select the degree of violation allowed for each constraint.

As can be observed the results (Tables 4 and 6) clearly show that the flexibilization achieved in case 1 Maxmin ($\alpha > 0.98$) givesbetter results than those of case 2 Maxmin ($\alpha > 0.94$). Of course these better results are achieved at the expense of having a much higher degree of satisfaction for the minimum membership value, respectively, case 1 = 0.94 and case 2 = 0.98.

4. Conclusion

In real projects, cost and duration of each activity are the subject of change during to the presence of many uncertainties such as inflation, economical and social stresses, execution errors of contractor, and natural events.

Our method solves the time-cost tradeoff problems under an uncertain environment and proved to be practicable through a real example of franchising implementation project performed by Serbian Postal Company.

The paper presents the results fuzzy GA algorithms and SA and their comparison with that of traditional approach for solving FLP problems.

A tradeoff analysis was performed both in terms of satisfaction of constraints and time to achieve results versus the objective function value. It should be noticed that, in general, if the decision maker intends to reach better constraints satisfaction level (i.e., coefficients and/or parameters) that will result in worse results in terms of the objective function and vice versa.

This study shows that the genetic algorithm produces better results than simulated annealing. The advantage of the GA method is that it can improve the solution using the coding function.

In real situations. depending on the project complexity, the shown models can help project managers while deciding which phases of the project allow a greater flexibility (the change of the time duration) and how those phases influence the costs. We leave for future work testing the model performance with an equivalent set of nonlinear problems.

In terms of future research it would be interesting to extend this study, through a complete fuzzification of the project time-cost tradeoff problem and to compare the results obtained with partial fuzzified problem to assess the possible gains of using our SA-GA approach as an algorithm.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The author thanks Professor Dr. Nebojša Bojović, and teaching assistant Dr. Miloš Milenković, from the Faculty of Transport and Traffic Engineering, University of Belgrade, for the suggestions in the work and enabled the data which were used in this research. The author acknowledges the financial support of the Ministry of Education, Science and Technological Development of the Republic of Serbia, within the Project no. III44006.

References

[1] L. J. Krajewski and L. P. Ritzman, *Operations Management: Process and Value Chains*, Prentice-Hill, Upper Saddle River, NJ, USA, 2005.

[2] A. H. Taha, *Operation Research: An Introduction*, Prentice Hall, New York, NY, USA, 2003.

[3] S. S. Leu, A. T. Chen, and C. H. Yang, "A GA-based fuzzy optimal model for constructiontime—cost trade-off," *International Journal of Project Management*, vol. 19, no. 1, pp. 47–58, 2001.

[4] C. Jin, Z. Ji, Y. Lin, Y. Zhao, and Z. Huang, "Research on the fully fuzzytimecost trade-off based on genetic algorithms," *Journal of Marine Science and Application*, vol. 4, pp. 18–23, 2005.

[5] E. Eshtehardian, A. Afshar, and R. Abbasnia, "Time-cost optimization: using GA and fuzzy sets theory for uncertainties in cost," *Construction Management and Economics*, vol. 26, no. 7, pp. 679–691, 2008.

[6] M. Ghazanfari, K. Shahanaghi, and A. Yousefli, "An application of possibility goal programming to the time-cost trade off problem," *Journal of Uncertain Systems*, vol. 2, pp. 22–28, 2008.

[7] M. Ghazanfari, A. Yousefli, M. S. Jabal Ameli, and A. Bozorgi-Amiri, "A new approach to solve time-cost trade-off problem with fuzzy decision variables," *International Journal of Advanced Manufacturing Technology*, vol. 42, no. 3-4, pp. 408–414, 2009.

[8] H. Ke, W. Ma, and J. Ma, "Solving project scheduling problem with the philosophy of fuzzy random programming," *Fuzzy Optimization and Decision Making*, vol. 11, no. 3, pp. 269–284, 2012.

[9] H. Ke, W. Ma, X. Gao, and W. Xu, "New fuzzy models for time-cost trade-off problem," *Fuzzy Optimization and Decision Making*, vol. 9, no. 2, pp. 219–231, 2010.

[10] S. Chen and M. Tsai, "Time-cost trade-off analysis of project networks in fuzzy environments," *European Journal of Operational Research*, vol. 212, no. 2, pp. 386–397, 2011.

[11] H. Ke and B. Liu, "Project scheduling problem with mixed uncertainty of randomness and fuzziness," *European Journal of Operational Research*, vol. 183, no. 1, pp. 135–147, 2007.

[12] H. Kwakernaak, "Fuzzy random variables. I. Definitions and theorems," *Information Sciences*, vol. 15, no. 1, pp. 1–29, 1978.

[13] H. Kwakernaak, "Fuzzy random variables.II. Algorithms and examples for the discrete case," *Information Sciences*, vol. 17, no. 3, pp. 253–278, 1979.

[14] F. M. Pires, J. M. Pires, and R. A. Ribeiro, "Solving fuzzy optimization problems: flexible appraches using simulated annealing," in *Proceedings of the World Automation Congress*, Monpellier, France, 1996.

[15] R. E. Bellman and L. A. Zadeh, "Decision-making in a fuzzy environment," *Management Science*, vol. 17, pp. 141–164, 1970.

[16] H. Zimmermann, *Fuzzy Set Theory and Its Applications*, Kluwer Academic, Dodrecht, The Netherlands, 1991.

[17] M. Fedrizzi, J. Kacprzyk, and J. L. Verfegay, "A survey of fuzzy optimization and mathematical programming," in *Interactive Fuzzy Optimization*, M. Fedrizzi, J. Kacprzyk, and M. Roubens, Eds., Springer, New York, NY, USA, 1991.

[18] Y. Lai and C. Hwang, *Fuzzy Multiple Objective Decision Making—Methods and Applications*, Springer, 1994.

[19] R. A. Ribeiro and F. M. Pires, "Fuzzy linear programming via simulated annealing," *Kybernetika*, vol. 35, no. 1, pp. 57–67, 1999.

[20] D. Dumitrecu, *Algoritmi genetici si strategii evolutive*, Editura Albastra, Cluj-Napoca, Romania, 2000.

[21] J. Filho, P. Treleaven, and C. Alippi, "Genetic-algorithm programming environments," *Computer*, vol. 27, no. 6, pp. 28–43, 1994.

[22] E. Aarts and J. Lenstra, *Local Search in Combinatorial Optimization*, John Wiley & Sons, 1997.

[23] L. R. Varela, R. A. Ribeiro, and F. M. Pires, "Simulated annealing fuzzy optimization," in *Proceedings of the 10th Mediterranean Conference on Control and Automation (MED '02)*, Lisbon, Portugal, July 2002.

[24] J. Blazewics, K. Ecker, E. Pesch, G. Schmidt, and J. Weglarz, *Scheduling Computer andManufacturing Processes*, Springer, Berlin, Germany, 1996.

[25] M. Milenkovic, N. Bojovic, R. A. Ribeiro, and N. Glisovic, "A Fuzzy Simulated Annealing approach for project time-cost tradeoff," *Journal of Intelligent and Fuzzy Systems*, vol. 23, no. 5, pp. 203–215, 2012.