Scheduling Construction Projects Using Evolutionary Algorithm

Piotr Jaśkowski[1] and Anna Sobotka[2]

Abstract: This paper attempts to use evolutionary algorithms to solve the problem of minimizing construction project duration in deterministic conditions, with in-time changeable and limited accessibility of renewable resources (workforce, machines, and equipment). Particular construction processes (with various levels of complexity) must be conducted in the established technological order and can be executed with different technological and organizational variants (different contractors, technologies, and ways of using resources). Such a description of realization conditions allows the method to also be applied to solving more complex problems that occur in construction practice (e.g., scheduling resources for a whole company, not only for a single project). The method's versatility distinguishes it from other approaches presented in numerous publications. To assess the solutions generated by the evolutionary algorithm, the writers worked a heuristic algorithm (for the allocation of resources and the calculation of the shortest project duration). The results obtained by means of this methodology seem to be similar to outcomes of other comparable methodologies. The proposed methodology (the model and the computer system) may be of great significance to the construction industry. The paper contains some examples of the practical use of the evolutionary algorithm for project planning with time constraints.

DOI: 10.1061/(ASCE)0733-9364(2006)132:8(861)

CE Database subject headings: Scheduling; Project management; Algorithms; Construction management.

Introduction

The most important part of construction project scheduling is the selection of resources (e.g., workforce, machines) and the harmonization of their work. Resource selection and harmonization are crucial for the success of a project for both the owner and the contractor. Each construction project is unique, therefore, the planner must take into account particular conditions (e.g., technological and organizational methods) and constraints (resource availability) to develop, the optimal project schedule, where optimal means the accepted evaluation criteria (e.g. project duration, cost, quality) are met. Contractors tend to give priority to the duration criterion (naturally on condition that the project budget is not exceeded), especially when:

- Acceleration of project execution may result in additional profits such as a bonus for finishing work ahead of time;
- Delays in project realization mean considerable losses;
- Cost of additional resources for accelerating construction is minor;
- There is a possibility of signing a more profitable contract; and
- Duration criterion is the crucial element taken into account by an investor when selecting and assessing bids.

Project duration is one of the main factors of competitiveness on the difficult construction market. Construction companies, especially Polish construction companies, have recently begun to reduce employment to reduce their fixed costs, but this reduction also limits their production capacity. Thus, construction project planning must be carried out by means of scheduling methods that allow for labor resource constraints, so that the construction project's duration can still be reduced and the contractors' standing and chances of obtaining an order despite limited resources can remain high. In practice, resource availability changes with time because contractors may engage resources in a number of parallel projects.

The main drawback of existing scheduling methods, especially precise and heuristic ones, is the fact that they fail to solve complex practical problems effectively and do not allow for real-world conditions and construction constraints. The writers present an approach that integrates the ideas of previous research based on evolutionary algorithms (the group of metaheuristic methods) to overcome the limitations of the existing methods.

Literature Review

The approaches to resource-constrained project scheduling can be summarized as follows:

1. Search for optimal solutions using integer programming, branch and bound techniques, dynamic and binary programming; and
2. Search for suboptimal solutions using heuristic algorithms, including:
    - Specialized heuristics; and

---

[1] Faculty of Civil and Sanitary Engineering, Lublin Univ. of Technology, ul. Nadbystrzycka 40, 20-618 Lublin, Poland. E-mail: piotr_jaskowski@wp.pl

[2] Associate Professor, Faculty of Mining and Geoengineering, AGH Univ. of Science and Technology, ul. Mickiewicza 30, 30-059 Krakow, Poland. E-mail: wsobotka@poczta.onet.pl

Note. Discussion open until January 1, 2007. Separate discussions must be submitted for individual papers. To extend the closing date by one month, a written request must be filed with the ASCE Managing Editor. The manuscript for this paper was submitted for review and possible publication on July 25, 2003; approved on January 3, 2006. This paper is part of the ***Journal of Construction Engineering and Management***, Vol. 132, No. 8, August 1, 2006. ©ASCE, ISSN 0733-9364/2006/8-861–870/$25.00.

- Metaheuristic methods—taboo search method, simulated annealing, genetic algorithms.

Artificial intelligence methods are also exploited in the form of expert systems, artificial neural networks, and hybrid systems.

The scheduling problem is often described as the task of integer programming. In such tasks the vector of decision variables usually takes the form of a binary vector (Brucker et al. 1999; Kasprowicz 2002; Kolish and Padman 1997; Marcinkowski 1990; Weglarz 1981). Precise procedures of single-criterion optimization of schedules are mainly based on the branch and bound method (Dorndorf et al. 2000; Kasprowicz 2002; Weglarz 1981).

Using precise algorithms to solve serious practical problems is impossible because of the length of time needed for the calculations and the limited memory capacity of computers (Marcinkowski 1990; Slowinski et al. 1994; Weglarz 1981). Thus several approximation methods employing the heuristic approach have been conceived. The methods can be divided into two groups: specialized heuristics and metaheuristics.

Specialized heuristics are used to solve only one optimization problem. Among the most popular heuristics solving scheduling problems is priority heuristics, which is available in the project-scheduling software in common use nowadays. Priority heuristics include two phases. In the first one, a so-called priority list, a list of processes, is prepared and arranged according to decreasing values of priority calculated on the basis of an assumed rule. In the second, the start and finish times of these processes are calculated so as to keep to all the constraints. In this phase, one of the two methods of tasks scheduling is used: parallel or serial, which differ in how they solve resources conflicts. Examples of the use of priority heuristics for scheduling projects and resources allocation can be found in the works of Shanmuganayagam (1989), Tsai and Chiu (1996), and Ulusoy and Özdamar (1995), among others. No certain priority rule produces the best results for certain problems and structures of projects. For this reason, Khamooshi (1996) has modified the existing approach to establish process priorities. The procedure Khamooshi worked out consists of dividing a project into parts and using a different priority rule for each part. He presents this approach in the form of a dynamic programming model. Slowinski et al. (1994) suggested employing a cluster of many rules instead of one priority rule, and then choosing the best one.

To solve single-criterion optimization problems in scheduling projects, metaheuristic algorithms can also be used. They define only a certain pattern of optimization procedure, which must be adapted for particular applications. The most frequently used metaheuristic methods are simulated annealing, taboo search method, and evolutionary algorithms. Both simulated annealing and taboo search method belong to the group of neighborhood (local) search algorithms (Sampson and Weiss 1993). They search the area of feasible solutions passing from a current solution to a neighboring one (the definition of "neighborhood" and the way of neighboring solutions generation depend upon the nature of the problem).

The idea of imitating processes that take place in nature used in local search methods is also used in evolutionary algorithms. Evolutionary algorithms include genetic algorithms, evolutionary programming, evolution strategies, classifier systems, genetic programming. The results of research in this field are usually not classified according to an individual method but are generally described as evolutionary algorithms (Michalewicz 1996).

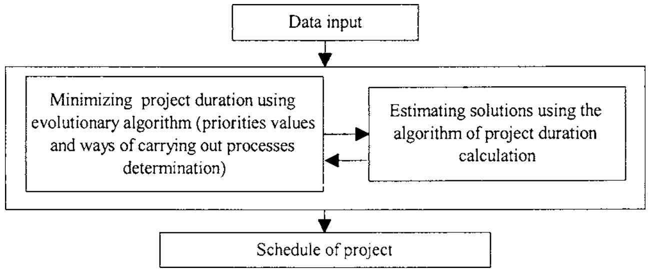

Fig. 1. Diagram of method of construction project scheduling

Evolutionary algorithms work as computer systems for solving problems according to the rules observed in the evolution of live organisms. The rules involve system structure and the organisms ways function and adapt to existing conditions. A feature of this approach to solving optimization problems is creating a population of individuals representing solutions in a form of a chromosome. As in nature, better-adapted individuals—or more effective solutions—stand a better chance of survival and development. The evolutionary algorithms are used to solve optimization problems in many branches of industry. A number of examples of their application, such as the optimization of structures (Koumousis and Georgiou 1994), engineering and design (Grierson and Pack 1993), and selection of equipment for earth-moving operations (Haidar et al. 1999), may be found construction. Current studies show that evolutionary algorithms have a considerable potential to solve many project scheduling problems efficiently. For example, Li et al. (Li and Love 1997; Li et al. 1999) use genetic algorithms to facilitate the time-cost optimization, and Hegazy (1999) applies them to the optimization of resource allocation and leveling. Leu and Yang (1999) developed a multicriteria optimization model for construction scheduling based on a genetic algorithm, which integrates the time-cost trade-off model, the resource-limited model, and the resource-leveling model.

Evolutionary algorithms are classified by many authors (Kolish and Padman 1997; Michalewicz 1996) as methods based on artificial intelligence, i.e., algorithms acting like human behavior. Apart from evolutionary algorithms, artificial neural networks (Adeli and Karim 1997), expert systems (Kanet and Adelsberger 1987) and hybrid systems (Li et al. 1999) are also employed for solving scheduling problems.

Because practical application of precise methods is limited by the complexity of practical problems and imperfection of heuristic methods, the writers search for optimal and suboptimal schedules for construction projects using evolutionary algorithms. This approach proved to be appropriate for solving scheduling problems and relatively simple in computation. Although the method proposed by the writers does not provide the optimal solution, the results are close to the optimum and can be obtained in a short time. Because evolutionary algorithms may be easily adapted to solving any type of problems, the proposed method is versatile and allows defining the case conditions and constraints freely.

Description of the Problem Solution Method

A schematic diagram of the proposed method for solving construction project scheduling problems is shown in Fig. 1. The schedule is the result of calculations found using the two algorithms described below. The evolutionary algorithm is used for searching the minimal project duration. The heuristic algorithm enables resources allocation and is used for calculation of the project duration.

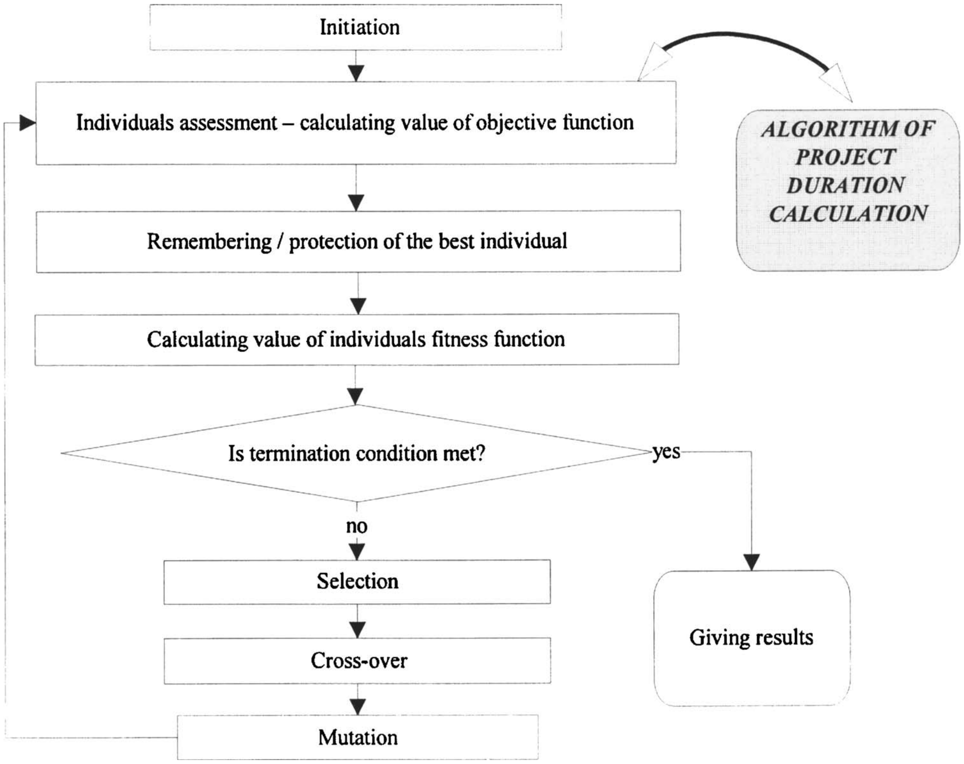

Fig. 2. Evolutionary algorithm

The Evolutionary Algorithm

The successive steps of the evolutionary algorithm are shown in Fig. 2 and described below. In Table 1, basic notions used in evolutionary algorithm description are explained.

Step 1. Initiation. Initiation consists in creating an initial population—a specified number of individuals (chromosomes). The writers use individual representation in the form of gene strings containing information about methods and values of processes priority (Fig. 3).

The initial population is created randomly. Particular genes assume values chosen randomly with equal probability from their variability interval.

Activity priorities, allocated randomly in the initial population, are modified in consecutive steps of the algorithm until a solution that corresponds to the shortest duration of the project is obtained. Therefore, the evolutionary algorithm enables the user to find optimal values of priorities that determine the sequence of allocating resources to activities. The algorithm is thus a tool that may help managers in their everyday work of making decisions and setting priorities.

Step 2. Individuals assessment. This procedure is used to calculate project duration, and thus it enables chromosomes (feasible solutions) assessment, ASSES[i]. To assess the solutions generated by evolutionary algorithm, the authors worked out the

Table 1. The Notions Used in the Evolutionary Algorithm Description

Notion	Explaining notions
Population	Set of individuals (solutions)
Individuals	Solutions encoded as chromosomes (strings of bits—genes, with information about ways of carrying out particular processes and processes priorities values)
Chromosomes	String of genes
Gene	Also called a feature, mark, detector, is a single element of a genotype (chromosome, in particular). In the study genes encode the method of carrying out a given process and the value of process priority
Genotype	A given individual's group of chromosomes
Phenotype	A set of values corresponding to a given genotype (time values for completing for a given solution)
Allel	Value of a given gene
Locus	Position of a given gene in a chromosome
Fitness function	The amount of adaptation (fitness) of a given individual in population; enables the selection of individuals best adapted in accordance with an evolutionary rule of survival of "the strongest"
Generation	A successive iteration in the evolutionary algorithm

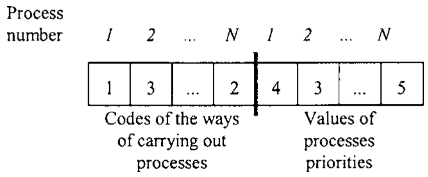

Fig. 3. Representation of feasible solutions (individuals) in form of chromosomes

heuristic algorithm for resources allocation and calculating the shortest project duration presented in the next section.

Step 3. Protection of the best individual. The individual (chromosome) from the initial population for which the objective function value is the best (the shortest project duration) is *remembered*. The *best individual protection* (so-called exclusive strategy) is a special additional reproductive procedure. The best adapted individual, among all from former generations, does not always pass to a new population. Exclusive strategy is used as the protective step against the loss of that individual. If the best individual from the current generation is worse than the best from the previous generation, then the latter replaces the worst individual in the current population.

Step 4. Calculating value of individuals fitness function. Evolutionary algorithms are used to look for the best adapted individuals for which the fitness function value is the highest. The study focuses on finding the solutions of minimization problems. In this case it is necessary to convert the minimized objective function into maximized fitness function FITNESS[i]:

$$\mathrm{FITNESS}[i]=\frac{\max\limits_{x}\mathrm{ASSES}[x]-\mathrm{ASSES}[i]+\mathrm{GAMMA}}{\max\limits_{x}\mathrm{ASSES}[x]-\min\limits_{x}\mathrm{ASSES}[x]+\mathrm{GAMMA}} \tag{1}$$

where $\min_x ASSES[x], \max_x ASSES[x]$=minimum and maximum value of the objective function in a given population; and GAMMA=factor that reduces differences among individuals in a population. The calibrating fitness function prevents premature convergence of the evolutionary algorithm, which would result in finding a local optimum and not a global one.

Step 5. Checking the termination condition. The action of the algorithm can be stopped in two cases: (1) after performing a specified number of iterations (when the number of the current generation is greater than the maximum value assumed), and (2) when, after some number of iterations, there are no better solutions than in previous generations. If the termination condition is not met, a selection of individuals is carried out as the next step.

Step 6. Selection procedure. Chromosomes selection (Fig. 4) consists in choosing individuals that will take part in producing offspring for the next generation. Chromosomes having the highest fitness function value are the most likely to produce new individuals. In the study, the roulette wheel method is used in the process of selection (Michalewicz 1996). Selection runs as follows:

- The entire fitness of a population is calculated as the sum SUM of fitness function values of all individuals;
- For each individual i, relative fitness FITNESSREL[i] is calculated corresponding to the probability of chromosome selection for reproduction:

$$\mathrm{FITNESSREL}[i]=\frac{\mathrm{FITNESS}[i]}{\mathrm{SUM}} \tag{2}$$

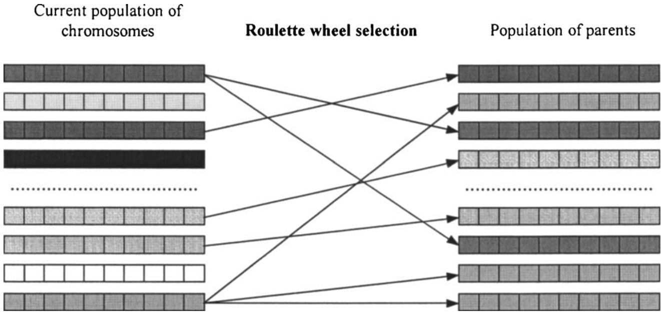

Fig. 4. Idea of selection procedure

- Then the total fitness FITNESSTOT[i] (cumulative distribution function of selection probability) is calculated by recurrent dependence:

$$\begin{gathered}\mathrm{FITNESSTOT}[0]=\mathrm{FITNESSREL}[0]\\ \mathrm{FITNESSTOT}[i]=\mathrm{FITNESSTOT}[i-1]+\mathrm{FITNESSREL}[i]\end{gathered} \tag{3}$$

- A random variable r within (0,1) is generated. An individual for which the condition

$$\begin{gathered}\mathrm{FITNESSTOT}[j-1]\le r<\mathrm{FITNESSTOT}[j]\\ (\text{individual } j=0, \text{ when:} \quad r<\mathrm{FITNESSTOT}[0])\end{gathered} \tag{4}$$

is met is selected for a new parental generation.

The last step is repeated for each individual in the population.

Step 7. Crossover. The task of crossover (Fig. 5) is to recombine chromosomes by exchanging strings of genes between parents' chromosomes. The one-point crossover was employed in the study. The procedure is carried out in two stages:

1. For each chromosome from the parental population, random variable y within (0,1) is generated. If y<PCROS where PCROS=crossover probability (system parameter), then a given chromosome is selected for recombination. Selected chromosomes are then paired.
2. For each pair of chromosomes (parents), a random number POINT is generated, defining the point of "crossing" chromosomes. Strings of genes in the parents' chromosomes ahead of the point of crossing are not changed, only genes behind that point are exchanged between parents.

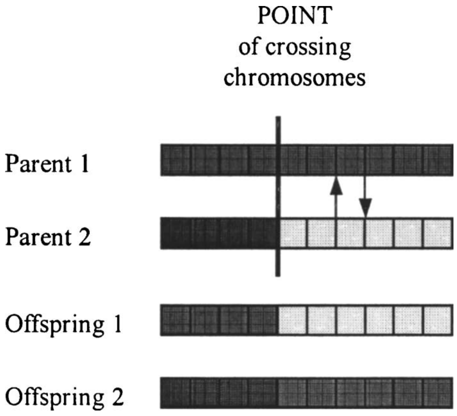

Fig. 5. One-point crossover

Step 8. Mutation. Mutation involves random change of one or more genes of the selected chromosome, with probability equal to mutation frequency PMUTAT. For each chromosome in a population and each gene in a chromosome, a random variable z within (0,1) is generated. If $z<$PMUTAT, a given gene j undergoes mutation, i.e., takes any value (selected randomly) within its variability interval.

Calculation of the fitness function value for each individual in a new generation, the best individual protection, selection procedures, crossover, and mutation are repeated cyclically until the termination condition of the algorithm is met. Then the result of algorithm's action, i.e., the solution to the problem—the way of using resources, the project duration, and the dates of beginning and finishing processes—is given. The best solution corresponds to the individual having the lowest value of the assessment function (the shortest or minimal project duration).

Algorithm of Individuals Assessment

The algorithm for calculating project duration is shown in Fig. 6 and consists of interactive allocation of renewable resources for processes and setting earliest completion dates resulting from resources availability and precedence relations. Resources are first allocated to the processes that have the highest priority. The algorithm establishes the shortest project duration for the methods and processes priorities defined (the information is encoded in the chromosome representing given possible solution).

Each successive step of the algorithm is explained as follows:

1. The variable defining the moment when resources will be allocated (TIME), and the finish dates TFPROC for all processes take zero value.
2. A set of processes (SET) is fixed. It comprises processes to which resources may be allocated at a given TIME moment.
3. Out of the SET, if it is not empty, the NRPROC process with the highest priority is chosen.
4. The availability of the renewable resources to carry out NRPROC process is checked. If resources are unavailable, NRPROC process is removed from SET and attached to SET2.
5. If resources are available, the start date of the process is settled for the TIME moment. The resources availability engaged in carrying out NRPROC process is changed.
6. The NRPROC process is removed from SET. Steps 3–6 are repeated for all the processes belonging to SET.
7. When no processes belong to SET, it is necessary to change the value of the TIME variable. The original procedure developed by authors is used here.
8. The algorithm is repeated from Step 2, and a new SET is created.
9. If all the processes have had their resources allocated, the date of completing project is calculated as the maximum deadline among all direct predecessors of the apex END.

Verification of Results

Verification of the results received from the developed computer system was carried out for ten test examples taken from PSPLIB library (Kolish and Padman 1997). These examples each include 12 to 16 processes. Each of them can be carried out in three ways with a different duration and demand for renewable resources. The verification results are shown in Table 2.

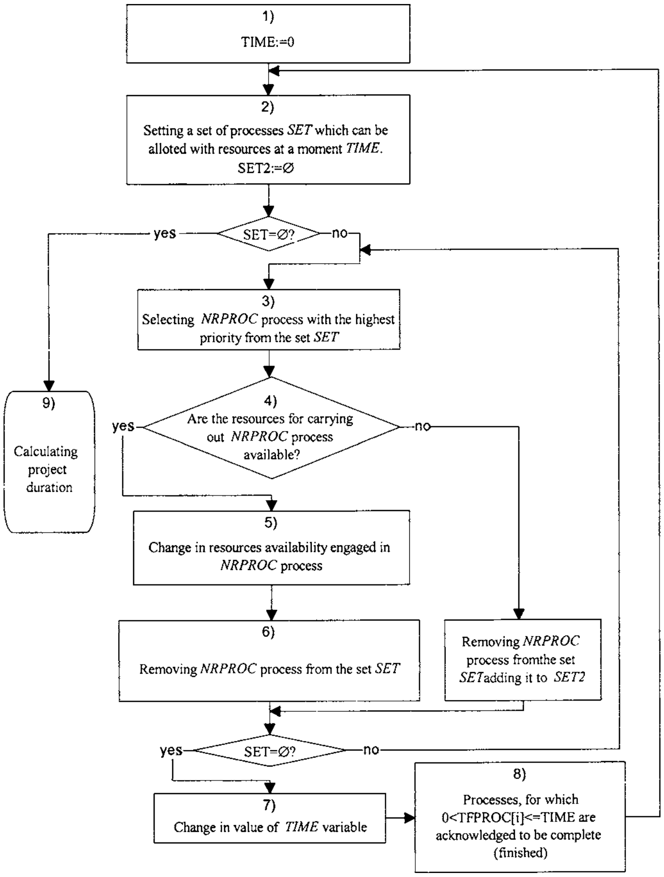

Fig. 6. Algorithm of the project duration calculation

For each example the best result, equal to optimum, was obtained. The solutions were found at the initial stage of the algorithm performance (the highest number of generations—59), i.e., after searching fewer than 2,950 feasible solutions at the most from among 43 mln possible, neglecting priority selection. Thus, the developed computer system quickly finds good quality solutions for selecting methods that demand limited resources.

All test examples examined so far concerned cases in which the availability of renewable resources is constant in time. Correcting the resource availability during the scheduling process (in accordance with Fig. 7, showing steps of the algorithm for calculating project duration) results in changes in number of available resources units in time. To confirm the correct performance of the algorithm, the authors used it to solve an example from the work by Marcinkowski (1990) concerning scheduling a construction project. The example (22 processes, 11 of them carried out using two methods, 2,048 variants altogether, 13 renewable resources) belongs to the class of multimode resources—constrained project scheduling problems. Additionally, some processes can only be carried out within definite time intervals.

To solve the problem, Marcinkowski uses a modification of Talbot's algorithm (Weglarz 1981). In consecutive iterations, time windows for each activity are calculated to conform to time and sequence constraints. Activities are scheduled based on applied priority rules. In case of conflicts, the Talbot's deductive review algorithm is used. Marcinkowski arrived at a solution for which project duration is 45 shifts (41 with unlimited resources availability).

Table 2. Calculation Results of Test Examples

Example	Optimum solution T_0 (days)	Number of processes	Number of possible variants of carrying out the project	Solution obtained by means of evolutionary algorithm T (days)	$(T-T_0)/T_0$ (%)
n010_1.mm	19	12	531,441	19	0.00
n010_10.mm	15	12	531,441	15	0.00
n010_2.mm	18	12	531,441	18	0.00
n013_4.mm	27	12	531,441	27	0.00
n016_3.mm	24	12	531,441	24	0.00
n019_9.mm	17	14	4,782,969	17	0.00
n022_2.mm	30	14	4,782,969	30	0.00
n024_10.mm	22	14	4,782,969	22	0.00
n028_1.mm	27	16	43,046,721	27	0.00
n030_5.mm	27	16	43,046,721	27	0.00

After introducing the additional resources necessary to complete processes with time constraints, available only during certain intervals, calculations were made using the designed computer system. The best solution—a project-duration 3 shifts shorter than in Marcinkowski (1990)—was obtained in the 56th generation.

The research proves that the developed system is highly useful in scheduling construction projects and can solve various practical problems.

Practical Application

Evolutionary algorithms have been used to find optimized solutions for many problems in different areas of economics for a long time (see literature review section). Investigations also confirm the utility of algorithms in construction project scheduling. This methodology allows for consideration of the different conditions and objectives of a construction project. In the present methodology (using evolutionary algorithms) developed by the authors, a user can freely define conditions and constraints that occur in practice; this is very important in the practical use of this method. This approach may be applied to solving problems with resources and time constraints in a whole project or its parts. These problems are very important when the production capacity of companies is limited. For example, in Poland over 97% of construction companies employ fewer than 20 workers. The present approach can be used to analyze the production capacity such Polish contractors and to verify if they can fulfill their contracts in time constraints established by the investor.

The computer system based on the developed algorithm may be a tool for solving various decision problems. Possible applications of the algorithm are illustrated by the three case studies presented below:

1. Scheduling when the start and finish dates of each work package are set in advance by the owner;
2. Global scheduling on the scale of a whole company, not only a single project; and
3. Updating the schedule during construction.

Practical Examples

To illustrate how the algorithm works, it was applied to plan the execution of a sports center consisting of a hall and a sports field and to design a set of machines owned by a contractor. Data for the example are shown in Tables 3 and 4, and in Fig. 7.

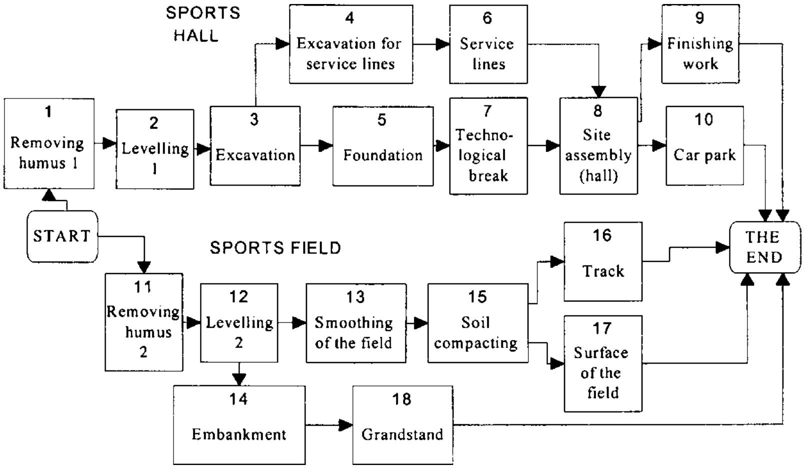

Fig. 7. Graph of technological dependencies among processes in example

Table 3. Available Number of Renewable Resources Units

Resource	Resource name	Time interval of resource availability (H)	Available number of resource units in time interval
R1	Bulldozer	(0, *H*)	2
R2	Loader	(0, *H*)	1
R3	Dump truck	(0, *H*)	3
R4	Road roller	(0, *H*)	2
R5	Haul scraper	(0, 30)	1
		(30, *H*)	0
R6	Excavator	(0, *H*)	1
R7	Excavator 0.15	(0, *H*)	1
R8	Crane	(0, 20)	1
		(20, *H*)	2
R9	Grader	(0, *H*)	1

Case Study

To simplify control over the project's execution and its budget, the owners demand that the contractors meet the deadlines for completing individual work packages that together make up the whole project. In the analyzed case study, the owner set a date of completion for the sports field (Processes 11 to 18 in Fig. 7) for the 40th day of the project, and the date of completion for the sports hall (Processes 1 to 10) for the 70th day. The prospective contractor had to analyze his capacities and resources (workforce and equipment) and develop a schedule of resource utilization that would meet the owner's requirements. As there are a number of organizational and technological alternatives for the particular processes, hence alternative utilization of resources, the result of the contractor's analysis should lead to a decision about which method to choose.

The algorithm developed by the authors allows the scheduler to model the constraints in the form of deadlines for the work packages. This can be done by introducing additional "dummy" resources into the model. In the analyzed case, the dummy resources were assumed for the Processes 16, 17, and 18, one unit per process. The availability of the dummy resources was assumed to be up to three units from the first to the 40th day of the construction. With these assumptions, the project duration

Table 4. Time of Carrying Out Processes and Demand for Renewable Resources

				Demand for resources (production units)								
Process number	Name of process	Method	Duration	R1	R2	R3	R4	R5	R6	R7	R8	R9
1	Removing humus 1	I	2	1	1	1	0	0	0	0	0	0
		II	1	2	1	1	0	0	0	0	0	0
2	Leveling 1	I	13	1	1	3	2	0	0	0	0	0
		II	9	0	0	0	0	1	0	0	0	0
3	Excavation	I	4	1	1	2	0	0	0	0	0	0
		II	2	0	0	3	0	0	1	0	0	0
4	Excavation for service lines	I	1	0	0	0	0	0	0	1	0	0
		II	4	0	0	0	0	0	0	0	0	0
5	Foundation	I	7	0	0	0	0	0	0	0	1	0
6	Service lines	I	4	0	0	0	0	0	0	0	0	0
7	Technological break	I	5	0	0	0	0	0	0	0	0	0
8	Site assembly of the hall	I	15	0	0	0	0	0	0	0	1	0
		II	8	0	0	0	0	0	0	0	2	0
9	Finishing works	I	28	0	0	0	0	0	0	0	0	0
10	Car park	I	14	1	0	1	2	0	0	0	0	1
11	Removing humus 2	I	2	1	1	1	0	0	0	0	0	0
		II	1	2	1	1	0	0	0	0	0	0
12	Leveling 2	I	12	1	1	2	0	0	0	0	0	0
		II	7	0	0	0	0	1	0	0	0	0
13	Smoothing the field	I	3	0	1	1	0	0	0	0	0	1
14	Embankment	I	5	1	0	1	2	0	0	0	0	0
		II	4	0	0	0	0	1	0	0	0	0
15	Soil compacting	I	5	0	0	0	2	0	0	0	0	0
16	Track	I	6	1	0	1	2	0	0	0	0	0
17	Surface of the field	I	10	0	0	0	0	0	0	0	0	0
18	Grandstand	I	10	0	0	0	0	0	0	0	1	0
		II	7	0	0	0	0	0	0	0	2	0

Table 5. The Impact of the Execution Time of the Sports Field on Minimal Project Duration (in the Example)

Availability of the dummy resource (days)	Execution time of the sports field (days)	Minimal project duration (days)
26–27	26	72
28–33	28	68
34	34	66
35–40	35	64
>40	26	60

corresponds to the duration of the hall construction. Changing the dummy resources' availability allows finding the relationship between the field's execution time and the shortest project duration (see Table 5).

The results of the calculations show that the contractor can adequately distribute the resources and complete the works in time. There are three possible schedules. Assuming that finishing ahead of time will result in an earlier payment, the optimal solution for the contractor was the one presented in Fig. 8, which allows the sports field to be completed on the 35th day and the total duration of the project to be 64 days. This is an "early start" schedule.

Further scheduling may include resources leveling or assuring continuity of resource use by means of moving the start dates of particular processes within the existing float time (reserves). This float time has been calculated on the basis of the evolutionary algorithm and is shown in Table 6. The duration of processes and their start and finish dates may be moved within the free float time without affecting other processes. Only Processes 7, 10, and 16 have free float time. The schedule is, therefore, vulnerable to change during the construction (see Case Study 3). In spite of that, the deadlines can be met, a result of the existing total float time. The total float time equals the maximum extension of process duration that does not cause the overrun of the work package deadline (40th day for the sports field and 70th day for the hall). The processes with the shortest total float time (1, 2, 12, 17 and 8, 9, 13, 18) may be declared subcritical (as in the critical path method); to know this is crucial for the project control.

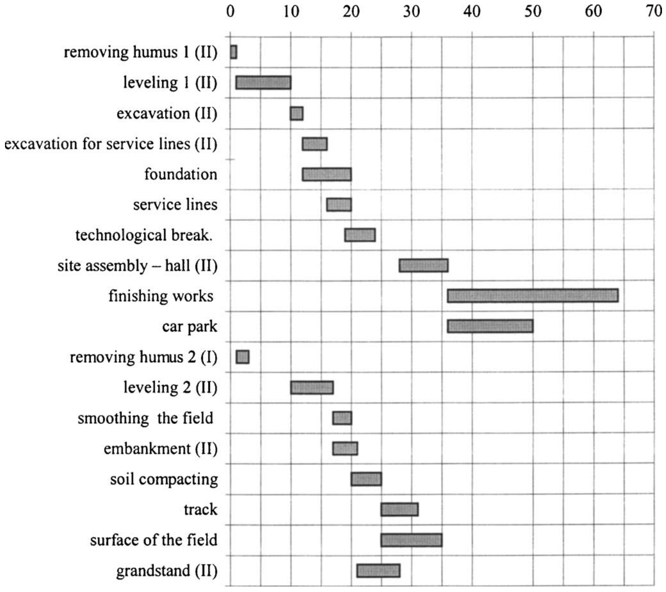

Fig. 8. Project schedule (finish date of sports field—35th day)

Table 6. Total and Free Float Time of Processes (in the Example)

Process number	Free float time (days)	Total float time (days)
1	0	5
2	0	5
3	0	8
4	0	14
5	0	8
6	0	14
7	4	10
8	0	6
9	0	6
10	14	20
11	0	12
12	0	5
13	0	6
14	0	9
15	0	13
16	4	9
17	0	5
18	0	6

Case Study 2

Contractors usually take on a number of parallel projects. Therefore, it is necessary to make allowance for parallel projects during new project scheduling. Resource availability can be determined on the basis of existing schedules, and for project scheduling, the methodology, presented in Case Study 1, can be used. Further, this problem may be considered as resource allocation on the scale of the whole company, including contracts, tasks, and processes. Obviously, scheduling on a company-wide scale allows all contractual deadlines to be met and the duration of current projects to be minimal, thus ensuring a company's competitiveness and economic resource use.

Implementing the methodology and the model described by the authors may also help to solve this problem. Let us assume that the contractor executes two independent projects for separate investors. The construction of the sports field has already been commissioned to the contractor (the date of completion of the sports field was set on the 40th day), and he is bidding for the sports hall. The start dates of each order (i.e., the field and the hall) are the same. One of the three possible schedules is thus as shown in Fig. 8. Different deadlines for starting the hall may be modeled by introducing additional dummy resources in the model.

Case Study 3

The schedule is a tool for controlling the project execution. Disturbances and contingencies such as a design change may alter any plans so that the schedule requires updating. The availability of resources may also change. Only prompt reactions and effective decisions allow keeping to the terms of a contract. This case illustrates how the method can be applied in schedule updating.

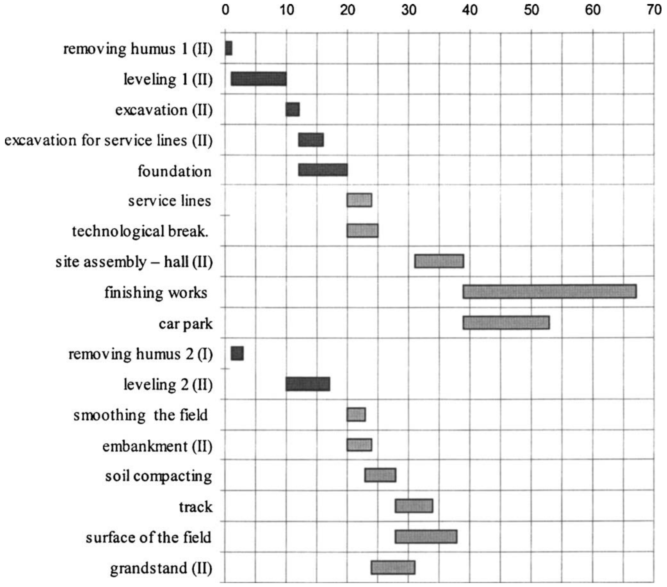

Fig. 9. Project schedule (finish date of sports field—38th day; project duration—67 days)

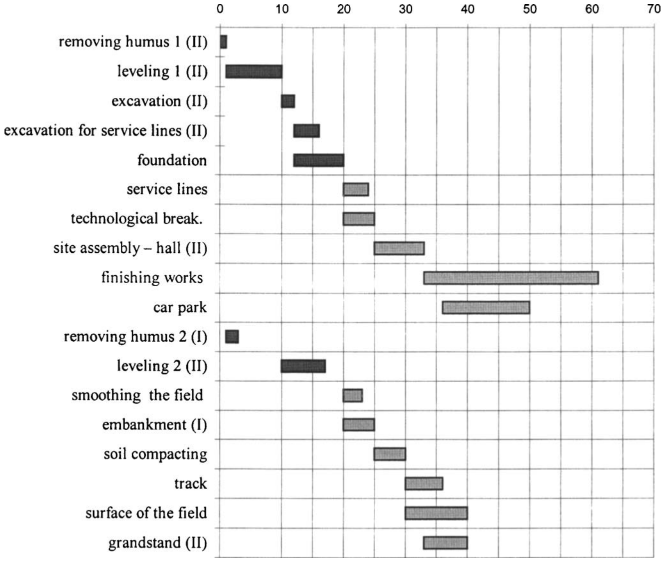

Fig. 10. Project schedule (finish date of sports field—40th day; project duration—61 days)

The project described in Case Study 1 has not been executed as planned. Processes 6, 7, 13, and 14 did not start until the 20th day of the project. The aim of schedule updating is to reschedule the allocation of resources so that the owner's deadlines can be met (i.e., completion of the field on the 40th day and finishing the hall on the 70th day). The new schedule is obtained by means of calculating the reconstructed project network, from which all accomplished processes have been removed. The new schedule show that it is possible to hand over the field on the 38th day and the hall on the 67th day at the earliest (Fig. 9). The organizational and technological alternatives for particular processes were in this case the same as in the original schedule.

The algorithm enables contractors to choose how the processes are executed. Therefore, a test was made to check if there was any better way of deploying resources to complete the project successfully. Another possible schedule that assumed the completion of the sports field on the 40th day and the hall on the day 61st day (Fig. 10) was also obtained. Thus, the developed algorithm facilitated finding a better solution, i.e., a schedule with an earlier project handover date.

Conclusion

Resource-constrained project scheduling problems can be solved using several methods. Because of the great computational complexity of precise methods, they are used for planning only a small number of processes. Practical solutions require making allowances for many different situations, conditions, and constraints in the planning process. The possibility of employing an evolutionary algorithm to solve practical problems was examined by many researchers. Existing methods overlook the limited production capacity of construction companies (contractors) that are simultaneously engaged in the realization of some projects. The paper examined the possibility of overcoming limitations in the practical use of existing methods. This aim was achieved through the description of the scheduling problem with resources accessibility limited and changeable in time. The verification research enabled the positive evaluation of the practical character of this approach, particularly in reference to the possibility of imitating complex conditions and solving many practical problems in construction company management. Possible practical applications were presented and analyzed in the attached case studies.

As the multiple execution mode options differ in terms of their total cost, the approach finds application in cases as described in the paper. In many practical situations the decision maker will search for time-cost tradeoffs. Therefore, the authors are working to develop the method further to allow for a number of optimization criteria of various significance. Other utility functions for assessing the fitness of solutions are being sought and methods of interactive bi-criteria optimization are being tested.

Notation

The following symbols are used in this paper:

ASSES[i] = assessment of feasible solution i;
FITNESS[i] = value of fitness function for solution i;
FITNESSREL[i] = relative fitness for individual i;
FITNESSTOT[i] = total fitness for individual i;
GAMMA = factor that lessens differences between individuals in population;
H = planning horizon;
N = number of processes in project;
NRPROC = process from set SET with highest priority;
PCROS = crossover probability;
PMUTAT = mutation probability;
POINT = point of crossing chromosomes;
r, y, z = random variables within (0,1);
SET = set of processes that can be allotted resources at given TIME moment;
SET2 = set of processes that demand for resources in given iteration exceeds their availability;
SUM = sum of fitness function values for all individuals in population;

T = project duration calculated by means of evolutionary algorithm: (days);
T_0 = minimum project duration (days);
$\mathrm{TFPROC}[i]$ = finish date of process i; and
TIME = variable defining moment when resources will be allocated.

References

Adeli, H., and Karim, A. (1997). "Scheduling/cost optimization and neural dynamics model for construction." *J. Constr. Eng. Manage.*, 123(4), 450–458.

Brucker, P., Drexl, A., Möhring, R., Neumann, K., and Pesch, E. (1999). "Resource-constrained project scheduling: Notation, classification, models, and methods." *Eur. J. Oper. Res.*, 112(1), 3–41.

Dorndorf, U., Pesch, E., and Phan-Huy, T. (2000). "A branch-and-bound algorithm for the resource-constrained project scheduling problem." *Math. Methods Oper. Res.*, 52(3), 413–439.

Grierson, D. E., and Pack, W. H. (1993). "Optimal sizing, geometrical and topological design using genetic algorithm." *Struct. Optim.*, 6(4), 151–159.

Haidar, A., Naoum, S., Howes, R., Tah, J. (1999). "Genetic algorithms application and testing for equipment selection." *J. Constr. Eng. Manage.*, 125(1), 32–38.

Hegazy, T. (1999). "Optimization of resource allocation and leveling using genetic algorithms." *J. Constr. Eng. Manage.*, 125(3), 167–175.

Kanet, J., and Adelsberger, H. (1987). "Expert systems in production scheduling." *Eur. J. Oper. Res.*, 29(1), 51–59.

Kasprowicz, T. (2002). *Construction project engineering*, Institute for Terotechnology in Radom, Warsaw, Poland (in Polish).

Khamooshi, H. (1996). "Network-based project planning and scheduling." *Ind. Manage. Data Syst.*, 96(8), 13–22.

Kolish, R., and Padman, R. (1997). "An integrated survey of project scheduling." *Technical Rep. 463, Manuskripten aus den Instituten für Betriebswirtschaftslehre der Universitat Kiel*, ⟨http://citeseer.ist.psu.edu/37700.html⟨ (Jan. 12, 2003).

Koumousis, V. K., and Georgiou, P. G. (1994). "Genetic algorithms in discrete optimization of steel truss roofs." *J. Comput. Civ. Eng.*, 8(3), 309–325.

Leu, S. S., and Yang, Ch.-H. (1999). "GA-based multicriteria optimal model for construction scheduling." *J. Constr. Eng. Manage.*, 125(6), 420–427.

Li, H., Cao, J. N., and Love, P. E. D. (1999). "Using machine learning and GA to solve time-cost trade-off problems." *J. Constr. Eng. Manage.*, 125(5), 347–353.

Li, H., and Love, P. E. D. (1997). "Using improved genetic algorithms to facilitate time-cost optimization." *J. Constr. Eng. Manage.*, 123(3), 233–237.

Marcinkowski, R. (1990). *Scheduling methods of construction project*, Military Univ. of Technology, Warsaw, Poland (in Polish).

Michalewicz, Z. (1996). *Genetic algorithms+data structures evolution program*, 3rd Ed., Springer, Berlin.

Sampson, S. E., and Weiss, E. N. (1993). "Local search techniques for the generalized resource constrained project scheduling problem." *Naval Res. Logistics Quart.*, 40(5), 665–676.

Shanmuganayagam, V. (1989). "Current float techniques for resources scheduling." *J. Constr. Eng. Manage.*, 115(3), 401–411.

Slowinski, R., Soniewski, B., and Weglarz, J. (1994). "A DSS for multi-objective project scheduling." *Eur. J. Oper. Res.*, 79(2), 220–229.

Tsai, D. M., and Chiu, H. N. (1996). "Two heuristics for scheduling multiple projects with resource constraints." *Constr. Manage. Econom.*, 14(4), 325–340.

Ulusoy, G., and Özdamar, L. (1995). "A heuristic scheduling algorithm for improving the duration and net present value of a project." *Int. J. Operat. Product. Manage.*, 15(1), 89–98.

Weglarz, J. (1981). *Control in system of operation-complex type*, Polish Scientific, Warsaw-Poznan, Poland (in Polish).