



## SCHEDULING CONSTRUCTION PROJECTS WITH RESOURCES ACCESSIBILITY LIMITED AND CHANGEABLE IN TIME

Piotr Jaśkowski<sup>1</sup>, Anna Sobotka<sup>2</sup>

<sup>1</sup>Doctor Eng, Faculty of Civil and Sanitary Engineering, Lublin University of Technology, ul. Nadbystrzycka 40, 20-618 Lublin, Poland. E-mail: piotr\_jaskowski@wp.pl

<sup>2</sup>Assoc Prof, Faculty of Civil and Sanitary Engineering, Lublin University of Technology, ul. Nadbystrzycka 40, 20-618 Lublin, Poland. E-mail: hanka@akropolis.pol.lublin.pl

Received 31 Aug 2004; accepted 28 Oct 2004

**Abstract.** This paper aims at solving the problem of minimising the construction project duration in deterministic conditions when the accessibility of renewable resources is limited and changeable in time (workforce, machines and equipment). Particular construction processes (with various levels of complexity) must be conducted in the established technological order and can be executed in different technological and organisational variants (different contractors, technologies, and ways of using resources). To solve this problem the authors are using evolutionary algorithm. For the assessment of solutions generated by evolutionary algorithm, the authors have worked out a heuristic algorithm (of resources allocation and project duration calculation). This methodology seems to produce similar outcomes when juxtaposed with other solutions obtained by research works carried out using comparable methodologies. The paper contains an example of practical application of evolutionary algorithm for construction project planning within resources and time constraints.

**Keywords:** project scheduling, time and resources constraints, evolutionary algorithm, metaheuristic methods, suboptimal solution, minimising project duration.

### 1. Introduction

The process of planning how to execute a construction project on every stage should take into account the existing conditions (eg feasible technological-organisational variants) and limitations (resources accessibility limits). As the result of planning, the optimal project schedule should be developed.

The approaches in resources constrained project scheduling can be divided into:

- a) searching for optimal solutions using integer programming, branch and bound technique, dynamic and binary programming;
- b) searching for suboptimal solutions using heuristic algorithms, including:
  - specialised heuristics,
  - metaheuristic methods – taboo search method, simulated annealing, genetic algorithms.

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

### 2. Literature review

The issue of scheduling is often described as the task of integer programming. In such tasks the vector of

decision variables usually takes the form of binary vector [1–6]. Precise procedures of single-criterion optimisation of schedules are mainly based on the branch and bound method [2, 7]. Branch and bound algorithm allowing to determine a set of active schedules was worked out, among others, by Dorndorf et al. An active schedule is one in which no process can begin earlier without violating technological or resource limitations, or delay another process. The algorithm takes account of limitations and uses them to reduce the area of search. As a result, a set of possible dates for starting processes and thereby a set of feasible solutions (only feasible solutions) is restricted.

Using precise algorithms to solve serious practical problems is impossible because of length of time needed for calculations and due to the limited memory capacity of the computer [4, 9, 10]. This has resulted in the formation of several approximation methods employing the heuristic approach. The methods can be divided into two groups: specialised heuristic and metaheuristic. Specialised heuristic methods are used to solve only one optimisation problem. In previous researches in the subject there are many examples for concrete problems and algorithms. For example, R. Marcinkowski [4] worked out a heuristic algorithm of project scheduling with various process technologies (*multi-mode*), time constraints

and existing limits of renewable resources availability. To solve the problem the author used B. F. Talbot's idea of algorithm [6]. M. Prystupa [5] worked out a heuristic algorithm for allocation of contractors and scheduling construction projects for situations where utility is maximised. He analyses the case where carrying out tasks will bring the investor profits which will depend on the time of their completion (usefulness is function of the completion date). This algorithm is based on iterative allocation of contractors for tasks.

Among the most popular heuristics solving scheduling problems are priority heuristics. These methods 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 priorities calculated on the basis of an assumed rule. In the other, the start and finish times of these processes are calculated so as to keep all the constraints. In this phase, one of the two methods of tasks scheduling is used: parallel or serial, which differ in the way of solving resources conflicts. In the serial methods [11–14] only one process is considered at a time, the one with the highest priority, in order to define its start time. If a process cannot be started at a given moment because of the lack of resources, then its execution is put off till the earliest time possible when resources are available. The procedure ends with the last element in the list. In a parallel method [9, 10] more processes are considered simultaneously. In the successive steps of the algorithm, a set can be defined including processes with the highest priorities and which can start at the moment because all their predecessors were carried out. There are no best priority rule producing best results for different problems and structures of projects. For this reason, H. Khamooshi [15] has modified the existing approach to establish process priorities. The procedure worked out by him consists in dividing a project into parts and using different priority rule for each part. He presents this approach in the form of a dynamic programming model. R. Slowinski, B. Soniewski and J. Weglarz [10] suggested employing a cluster of many rules instead of one priority rule, and then choosing the best one.

To solve single-criterion optimisation project scheduling problems, metaheuristic algorithms are used as well. They define only a certain pattern of optimisation procedure which must be adapted for particular applications. The most frequently used metaheuristic methods are as follows: simulated annealing, taboo search and evolutionary algorithms. Both simulated annealing and taboo search methods belong to the group of neighbourhood local search algorithms [16]. They search the area of feasible solutions passing from a current solution to a neighbouring one (the definition of „neighbourhood” and a way of neighbouring solutions generation, depend upon the nature of the problem). In taboo search method, the notion of neighbourhood refers to a given solution and to the algorithm step. In successive iterations the

best solutions found last are removed from neighbourhood. Such solutions are called taboo solutions, ie forbidden in next iterations. It makes easier to find global optimum and eliminate the area of local optimum. This approach also limits cycles in algorithm performance. In simulated annealing algorithms, if the neighbouring solution is better than a current one, it becomes a new current solution in the successive iteration. If it is worse, it can become a current solution in the successive iteration with certain probability dependent on the deterioration value of criterion function.

The idea of imitating processes taking place in nature used in local search methods, is also used in evolutionary algorithms. Evolutionary algorithms work as computer systems for solving problems according to rules observed in the evolution of live organisms. The rules involve system structure, and ways of their functioning and adapting to existing conditions. A characteristic feature of this approach in solving optimisation problems is creating a population of individuals representing solutions in a form of chromosomes. As in nature, better adapted individuals – solutions better from the point of view of an objective function – stand a better chance of survival and development. Some examples of employing evolutionary algorithms for solving scheduling projects in industry (both in problems of job-shop type and network planning) were worked out by P. Husbands [17]. H. Li et al [18] use improved genetic algorithm to facilitate time-cost optimisation. M. Pawlak [11] uses evolutionary algorithm for resources levelling in scheduling of industrial production. He employed direct solutions in the form of chromosomes in which successive genes mark the beginning data of carrying out a process. It required working out a procedure repairing solutions (chromosomes) which would meet model limiting conditions (order dependencies between processes).

Evolutionary algorithms are classified by many authors [3, 7, 17, 19] as methods based on artificial intelligence, ie algorithms acting like human behaviour. Apart from evolutionary algorithms, there are also artificial neural networks [20], expert systems [21] and hybrid systems [19], using evolutionary algorithms employed for solving scheduling problems.

Because of the fact that practical application of accurate methods of project scheduling is limited (high level of complexity of the practical problems), and due to imperfection of heuristic methods, the authors search in the present study for the suboptimal schedules of construction projects basing on evolutionary algorithms. The method proposed by the authors does not provide the optimal solution, the results are close to the optimum and obtained in a short computation time. Because the 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.

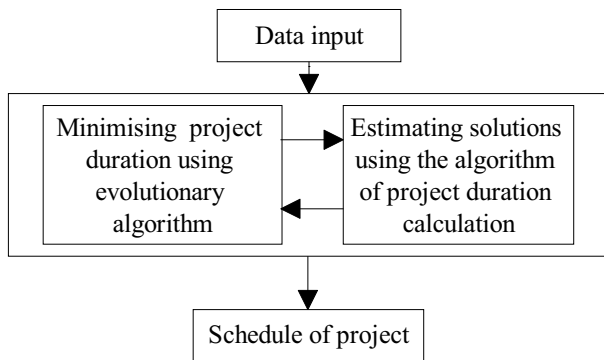


Fig 1. Diagram of the method of construction project scheduling

### 3. Description of a problem solution method

A schematic diagram of the proposed method of solving construction project scheduling problems is shown in Fig 1.

The project schedule is the result of calculations carried out by means of two algorithms described below. The evolutionary algorithm is used for searching the minimal project duration. The heuristic algorithm enables resource allocation for which the accessibility is limited and changeable in time and project duration calculation. Successive steps of evolutionary algorithm are shown in Fig 2. In Table 1 basic notions used in describing evolutionary algorithms are explained.

*Initiation* consists in creating initial population – a specified number of individuals (chromosomes). In the article the authors used individuals’ representation (feasible solutions) in the form of genes’ string containing information about ways of carrying out processes and values of priorities (Fig 3).

Table 1. The notions used in evolutionary algorithm description

Notion	Explaining notions
Population	Set of individuals (solutions)
Individuals	Solutions encoded as chromosomes (strings of bits with information about ways of carrying out processes and processes priorities values)
Chromosomes	String of genes
Gene	Also called a feature, mark, detector; in the study genes encode ways of carrying out a given process or/and value of process priority depending on placing the gene in a chromosome
Fitness function	It is the amount of adaptation of a given individual in population; it enables the selection of individuals best adapted in accordance with an evolutionary rule of surviving „the strongest”
Generation	A successive iteration in the algorithm

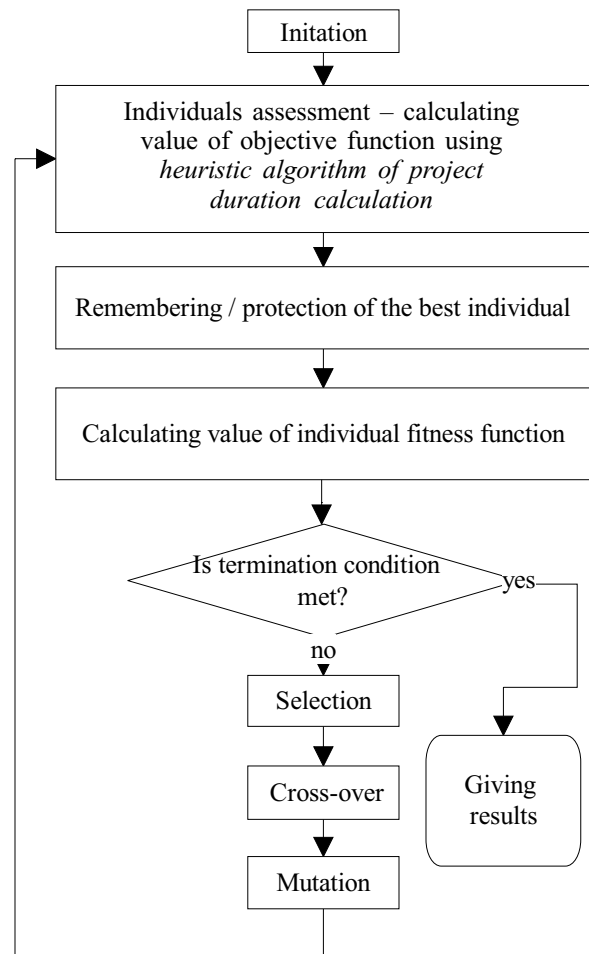


Fig 2. Evolutionary algorithm

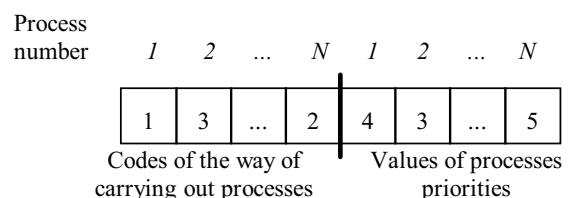


Fig 3. Representation of feasible solutions (individuals) in the form of chromosomes

In the assumed way of individuals’ representation, the number of process is identified through the position of a gene in chromosome.

Initial population is created randomly. The process consists of creating a number of chromosomes defined in advance. Particular genes assume values chosen randomly with equal probability from their variability interval. The procedure *individual assessment* is used to calculate the project duration for each individual of current population and thus it enables chromosomes assessment  $ASSES[i]$ . The *algorithm of project duration calculation* is shown in Fig 4. It consists in interactive allocation of renewable resources for processes and in setting dates

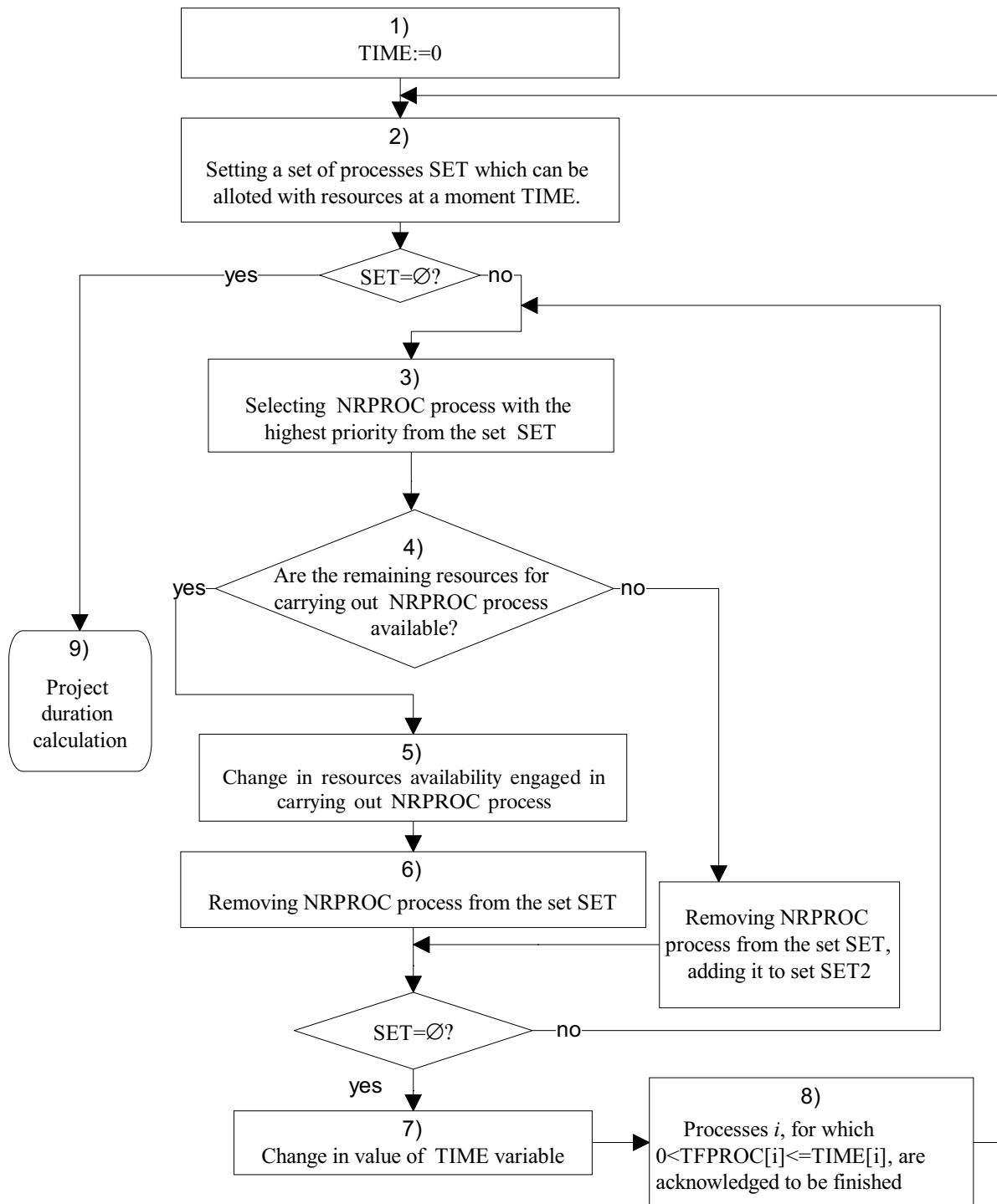


Fig 4. Algorithm of project duration calculation

for their completion at the earliest moments resulting from meeting the condition of resources availability and precedence relations among processes. Resources are first allocated to these processes that have the highest priority. The algorithm establishes the shortest project duration with defined ways of carrying processes and processes priorities (the information is encoded in the chromosome representing a given feasible solution).

Each successive step of the algorithm is explained as follows:

1. The variable defining the moment when resources will be allocated ( $TIME$ ) takes zero value.
2. A set  $SET$  of processes which can be allotted resources at a given moment  $TIME$  is fixed. The processes must meet two conditions:
  - they will not begin until the moment  $TIME$ ,
  - all the predecessors of each process were finished (at the moment  $TIME$  at the latest).
3. Out of the set  $SET$ , if it is not empty, the  $NRPROC$  process with the highest priority is chosen. In case

- of several processes having the same value of priority, the one having the smallest number is chosen.
4. Checking the availability of renewable resources indispensable in carrying out *NRPROC* process. If any resource is unavailable, *NRROC* process is removed from set *SET* and attached to *SET2* set. *SET2* contains processes whose demand for resources in a given iteration exceeds their availability.
  5. If resources are available, the start time for the process *NRPROC* is settled for a moment *TIME*. The resources availability engaged in carrying out *NRROC* process is changed. The process consists in creating a new availability interval or intervals, or requires correction of the number of resources units available in one or several time intervals.
  6. The process *NRPROC* which was allotted resources, is removed from *SET* set. The start time of the process is settled for a moment *TIME*. Steps from 3–6 are repeated for all the processes belonging to set *SET*.
  7. As there are no processes belonging to set *SET*, it is necessary to change the value of *TIME* variable. There An original procedure is used developed by the authors. A new moment of time when resources will be allotted, is defined in the following way:
    - searching the earliest finish time *TI* ( $TI > TIME$ ) among processes which were allotted resources,
    - for processes from *SET2* set, the earliest *H* date is looked for ( $TIME < H < TI$ ) in which the number of resources available enables us to begin carrying out any process. If *SET2* is empty, *TIME* variable takes *TI* value, otherwise the smallest value of *H* or *TI* is taken. The suggested procedure enables shortening the time of completing a construction project when resources availability is either constant or changeable in time.
  8. Processes  $[i]$  which meet the condition  $0 < TFPROC[i] \leq TIME$  are acknowledged to be finished. The algorithm is repeated from step 2 and a new set *SET* is marked.
  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*.

The individual (chromosome) from the initial population for which the objective function value is the best (the shortest project duration) is *remembered*.

Evolutionary algorithms are used to look for the best adapted individuals for which the fitness function value is the highest. In the study it is necessary to convert minimised objective function into maximised *fitness function*:

$$FITNESS[i] = \frac{\max_x ASSES[x] - ASSES[i] + GAMMA}{\max_x ASSES[x] - \min_x ASSES[x] + GAMMA}, \quad (1)$$

where:  $\min_x ASSES[x], \max_x ASSES[x]$  – minimum and maximum value of objective function in a given population,

*GAMMA* – factor which allows to reduce differences among individuals in a population (calibrating fitness function) and ensures that denominator in formula (1) does not equal zero.

Calibrating fitness function prevents premature convergence of evolutionary algorithm, which would result in finding a local optimum and not a global one.

The action of algorithm can be stopped in two cases (*the termination conditions*):

- after performing a specified number of iterations (when the number of current generation is greater than the maximum value assumed),
- 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. *Chromosome selection* consists in choosing these individuals that will take part in producing offsprings for the next generation. Chromosomes having the highest value of fitness function are the most likely to produce new individuals. In the study the method of roulette wheel is used in selecting procedure [22]. Selection runs as follows:

- entire fitness *SUM* of population is calculated as the 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

$$FITNESSREL[i] = \frac{FITNESS[i]}{SUM}, \quad (2)$$

- and then total fitness  $FITNESSTOT[i]$  (cumulative distribution function of selection probability) by recurrent dependence:

$$\begin{aligned} FITNESSTOT[0] &= FITNESSREL[0] \\ FITNESSTOT[i] &= FITNESSTOT[i-1] + FITNESSREL[i], \end{aligned} \quad (3)$$

- a random number *r* within is generated. An individual for which the condition

$$FITNESSTOT[j-1] \leq r < FITNESSTOT[j] \quad (4)$$

(for individual  $j = 0$ , when  $r < FITNESSTOT[0]$ )

is met, is selected for a new parental generation.

The last step is repeated as many times as the population size. Because of that, in the selection process the parental population is of the same size as the current population. In the population some individuals (better adapted) can appear many times.

The task of *crossover* 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 parental population a random number *x* within  $(0,1)$  is generated. If

$x < PCROSS$  where PCROSS is 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 is generated, defining the point of "crossing" chromosomes. Strings of genes in parents' chromosomes ahead of the point of crossing are not changed, only genes behind that point are exchanged between parents.

New individuals created in this way are included in a new generation, subject to *mutation* operator at a further stage. Mutation consists in random change of one or more genes of the selected chromosome, with probability equal to mutation frequency *PMUTAT*. In the study, regular mutation was used in which probability of mutation of each gene is identical, and operator's action does not depend on the number ("age") of a generation. For each gene in a chromosome, a random number  $y$  within  $(0,1)$  is generated. If  $y < PMUTAT$ , a given gene  $j$  undergoes, mutation ie takes any value (selected randomly) within its variability interval.

The *best individual protection* (so-called exclusive strategy) is a special additional reproductive procedure. The best adapted individual among all former generations, does not always passes to a new population. Exclusive strategy is used as the protective step against the loss of that individual - it is included into a new population. 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.

Calculation of the fitness function value for each individual in a new generation, the best individual protection, selection, crossover and mutation procedures are repeated cyclically until the termination condition of the algorithm is met. Then the result of algorithm's action is

given, ie the solution to the problem - the way of using resources and carrying out particular processes, the shortest project duration and, for the best solution - start and finish times of each process. The best solution corresponds to the individual having the best (lowest) value of the assessment function (the shortest or minimal project duration).

#### 4. Verification of results

Verification of results received from the developed computer system was carried out using ten test examples taken from PSPLIB library [3]. These examples include from 12 to 16 processes. Each of them can be carried out in three ways with a different duration and with a different demand for renewable resources. The verification results are shown in Table 2.

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 generation – 59), ie after searching less than 2950 feasible solutions at the most from among 43 mln possible – neglecting priority selection. Thus the developed computer system finds quickly good quality solutions for selecting ways of carrying out processes with a demand for limited resources.

All test examples examined so far concerned to the case when available number of renewable resources is constant in time. The necessity to correct the resources availability during the scheduling process results in changes in number of available resources units in time. In order to confirm the correct performance of the algorithm, the authors used it to solve an example from [4] concerning scheduling a construction project. The example (22 processes, 11 of them can be carried out using two methods, 2048 variants altogether, 13 renewable

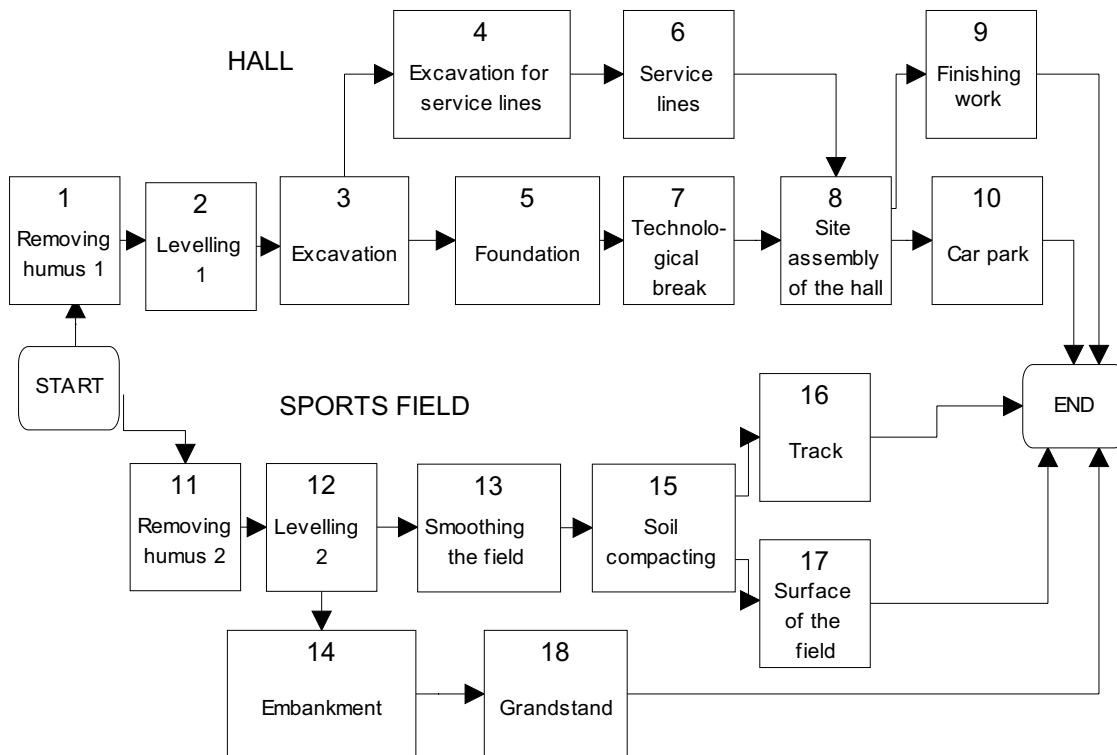
**Table 2.** Calculation results of test examples

Example marking (according to [10])	Optimum solution $T_0$ [days] (according to [10])	Number of processes	Number of possible variants of carrying out project	The number of generation with the best solution obtained	Solution obtained by means of evolutionary algorithm $T$ [days]	$(T - T_0) / T_0$ [%]
1	2	3	4	5	6	7
n010_1.mm	19	12	531441	7	19	0.00
n010_10.mm	15	12	531441	25	15	0.00
n010_2.mm	18	12	531441	28	18	0.00
n013_4.mm	27	12	531441	18	27	0.00
n016_3.mm	24	12	531441	5	24	0.00
n019_9.mm	17	14	4782969	20	17	0.00
n022_2.mm	30	14	4782969	21	30	0.00
n024_10.mm	22	14	4782969	3	22	0.00
n028_1.mm	27	16	43046721	9	27	0.00
n030_5.mm	27	16	43046721	59	27	0.00



**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]																Unlimited resources	
	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40		
Execution time of the sports field	–	26		28						34		35						26
Minimal project duration	–	72		68						66		64						60



**Fig 5.** Graph of technological dependences among processes in the example

tractor had to analyse his capacities and resources and develop a schedule of resources utilisation that would meet the owner’s requirements. 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 of certain availability (engaged in carrying out processes: 16, 17 and 18; one unit per process). The availability of the dummy resources was assumed to be up to 3 units from the first to the 40<sup>th</sup> day of the construction. With these assumptions, the execution time of the whole project corresponds to the execution time of the hall. Changing the dummy resources availability allows to find the relationship between the field execution time and the shortest project duration (Table 5). The results of calculations show that the contractor disposes of adequate resources and is able to 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 6, where the sports field is completed on day 35 and the total duration of the project is 64 days. This is an „early start” schedule.

Analysing the schedule in Fig 6, it can be ascertained that several variants for using disposable resources can correspond to one duration time of the project, for instance, removing humus 2 can be done with a haul scraper and not a bulldozer, it will not influence the final date. Selection of the schedule to be carried out requires employing additional criteria, eg cost. As the schedule is concerned with the earliest starting dates of processes, at a further stage of planning one can aim at compensating resources or maintaining continuity of their work, changing dates within workable time reserve.



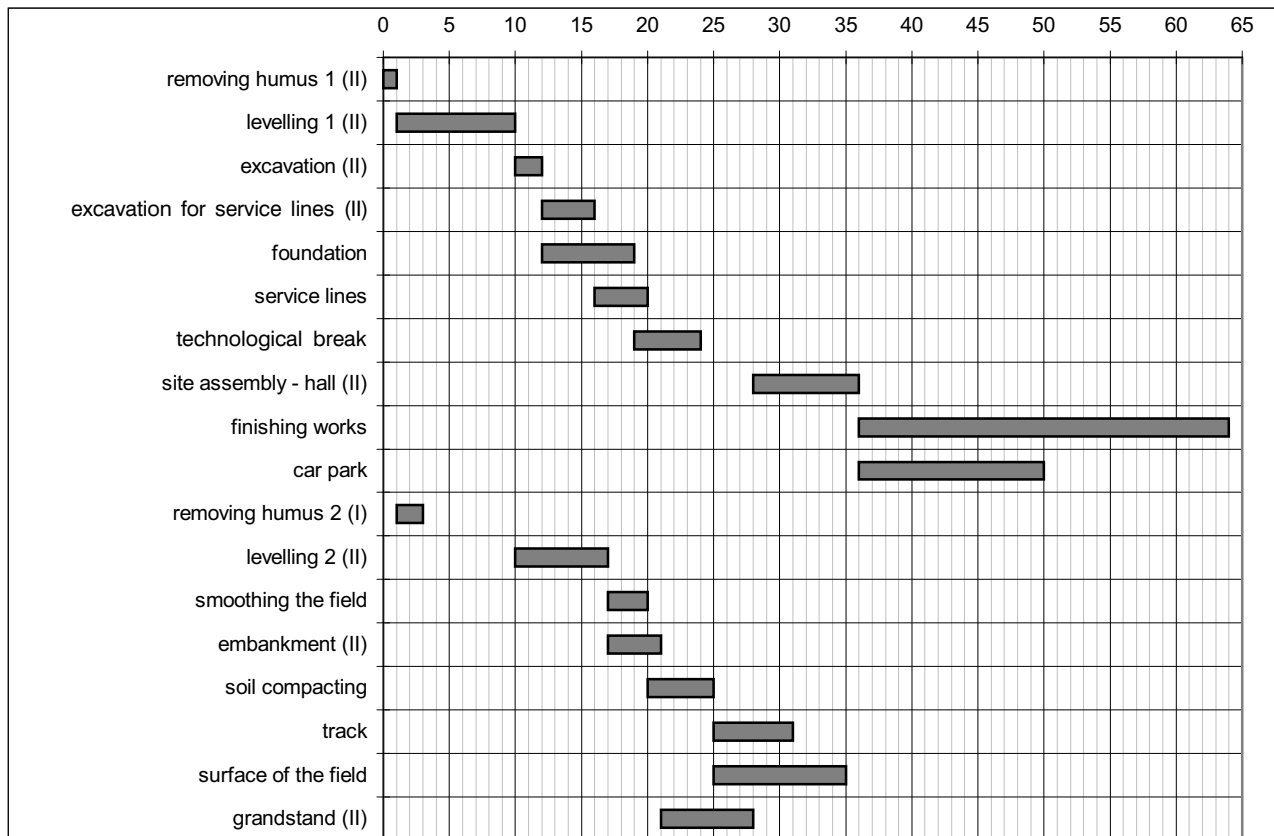


Fig 6. The project schedule (finish date of the sports field – 35th day; project duration – 64 days)

## 6. Conclusions

Resources constrained project scheduling problems are solved using several methods. In view of great computation complexity of precise methods, they are used only for scheduling test problems with a small number of processes and require simplifying assumptions. Practical problems require making allowances for many different situations, conditions and limitations in the planning process. In practical uses metaheuristic algorithms are more appropriate methods. Possibility of employing evolutionary algorithm to solve these problems was examined in the present article. Verification research carried out enabled to evaluate positively practical character of this method, particularly in reference to the possibility of imitating complex conditions of carrying out a construction project.

## References

1. Kasproicz, T. Construction project engineering. ITE Publisher in Radom (www.itec.radom.pl), Warsaw 2002 (in Polish).
2. Dorndorf, U.; Pesch, E. and Phan-Huy, T. A branch-and-bound algorithm for the resource-constrained project scheduling problem. *Mathematical Methods of Operations Research*, 52, 2000, p. 413–439.
3. Słowiński, R.; Soniewski, B. and Węglarz, J. A DSS for multiobjective project scheduling. *European Journal of Operational Research*, 79, 1994, p. 220–229.
4. Tsai, D. M. and Chiu, H. N. Two heuristics for scheduling multiple projects with resource constraints. *Construction Management and Economics*, 14, 1996.
5. Sampson, S. E. and Weiss, E. N. Local search techniques for the generalised resource constrained project scheduling problem. *Naval Research Logistics*, 40, 1993, p. 665–675.
6. Michalewicz, Z. Data Structure, Genetic Algorithms (Evolutionary Programs). WNT, Warsaw, 1999 (in Polish).
7. Węglarz, J. Control in System of Operation–Complex Type. Polish Scientific Publishers, Warsaw–Poznan, 1981 (in Polish).
8. Kolish, R. and Padman, R. Integrated survey of project scheduling. Institut für Betriebswirtschaftslehre Christian-Albrechts-Universität Kiel, Carnegie Mellon University Pittsburgh, <http://citeseer.nj.nec.com/37700.html>. 2000.
9. Marcinkowski, R. Scheduling Methods of Construction Project. WAT Publishers, Warsaw, 1990 (in Polish).
10. Li, H.; Cao, J. N. and Love, P. E. D. Using machine learning and GA to solve time-cost trade-off problems. *Journal of Construction Engineering and Management*, ASCE, September/October 1999.
11. Khamooshi, H. Network-based project planning and scheduling. *Industrial Management & Data Systems*, 96/8, 1996, p. 13–22.

12. Li, H.; Cao, J. N. and Love, P. E. D. Using improved genetic algorithms to facilitate time-cost optimisation. *Journal of Construction Engineering and Management*, ASCE, September 1997.
13. Adeli, H. and Karim, A. Scheduling / Cost optimisation and neural dynamics model for construction. *Journal of Construction Engineering and Management*, ASCE, December 1997.
14. Kanet, J. and Adelsberger, H. Expert systems in production scheduling. *European Journal of Operational Research*, 29, 1987, p. 51–57.
15. Hapke, M. Fuzzylogic multi-criteria network programming. Doctoral thesis. Information technology Institute. Technical University of Poznan, 1998 (in Polish).
16. Husbands, P. Genetic algorithms for scheduling. *AISB Quarterly*, 89, 1994. <http://www.cogs.susx.ac.uk>.
17. Prystupa, M. Allocation Models in Construction Processes. Warsaw University of Technology Publishers, Warsaw, 1988 (in Polish).
18. Pawlak, M. Evolutionary Algorithms as Tool In Production Scheduling. Polish Scientific Publishers, Warsaw 1999 (in Polish).
19. Shanmuganayagam, V. Current float techniques for resources scheduling. *Journal of Construction Engineering and Management*, 117, 1989.
20. Brucker, P.; Drexel, A.; Möhring, R.; Neumann, K. and Pesch, E. Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, 112, 1999, p. 3–41.
21. Jaworski, K. M. Construction design methodology. Polish Scientific Publisher PWN, Warsaw 1999 (in Polish).
22. Ulusoy, G. and Özdamar, L. A heuristic scheduling algorithm for improving the duration and net present value of a project. *International Journal of Operations & Production Management*, 15(1), 1995, p. 89–98.

### Appendix. Notation

*ASSES*[*i*] – value of the objective function for the feasible solution *i*;

*FITNESS*[*i*] – value of the fitness function for the solution *i*;

*FITNESSTOT*[*i*] – total fitness for individual *i*;

*FITNESSREL*[*i*] – relative fitness for individual *i*;

*GAMMA* – factor which allows to reduce differences between individuals in a population;

*H* – the earliest date in which the available number of resources enables to begin carrying out any of the process from the set *SET2*;

*NRPROC* – process from the set *SET* with the highest priority;

*PCROSS* – crossover probability

*PMUTAT* – mutation probability;

*SET* – set of processes which can be allotted resources at a given *TIME* moment;

*SET2* – set of processes whose demand for resources in a given iteration exceeds their availability;

*SUM* – sum of fitness function values for all individuals in population;

*TFPROC*[*i*] – finish time of process *i*;

*TI* – the earliest finish time among processes which where allotted resources and for which  $TI > TIME$ ;

*TIME* – variable defining the moment when resources will be allocated.