

## THE CONCEPT OF GENETIC PROGRAMMING IN ORGANIZING INTERNAL TRANSPORT PROCESSES

Konrad Lewczuk

Warsaw University of Technology, Faculty of Transport, Warsaw, Poland  
e-mail: kle@wt.pw.edu.pl

---

**Abstract:** *The paper presents proposition of using genetic algorithm to support organization of internal transport processes in logistics facilities. The organization of internal transport can be done through solving optimization task of scheduling internal transport process with allocation of human resources and equipment to the tasks. Internal transport process was defined and discussed as an object of organization. Precise methods of solving proposed optimization task were unable to give useful solutions according to the computational complexity of the problem, so heuristic genetic algorithm was proposed. The possible structures of chromosome representing feasible solutions, methods of generating initial population, base genetic operators: selection and inheritance, crossover, mutation and fixing of individuals were described. The main implementation difficulties, computational experiments and the scope of application of the algorithm were discussed.*

**Key words:** *scheduling, internal transport process, optimization, genetic algorithm.*

---

### 1. Introduction

Internal transport is a substantial part of logistics processes performed in all types of logistics facilities. Proper Organization of Internal Transport (ITO) is essential for effectivity and reliability of nodal elements of logistics systems and can be realized through scheduling (Ambroziak, 2007). Scheduling theory is widely discussed in the literature, but the area of internal transport is not clearly undertaken.

Internal transport system carries material flows from the facility entrance, by buffering (storage) areas, to the exit. Internal transport system realizes transportation process defined as set of sequenced handling tasks performed on materials in order to move and convert them up to the work-orders. It is realized by resources: people and equipment (transport devices). Tasks of the process emerge from technological sequence to which all handled materials are submitted. Tasks are composed of daily repetitive short transportation cycles or operations. The numbers of repetitions result from daily workload resulted from structure of shipments and receiving. Tasks can be performed parallel and can be divided into stages up to the work schedule. Typically, randomness of daily work load is covered by multiplying average daily flow volumes by increasing coefficient.

Fijałkowski (2003, 2012) formulates the issue of scheduling warehousing process (and indirectly ITO) with regard to labour consumption and costs of work. On that base Ambroziak and Lewczuk (2008a, 2008b) developed a mathematical model of scheduling warehousing process with elements of space optimization. Then Lewczuk (2011) continued it and set scheduling rules in application to warehouse design. Also other authors undertake the problem. Kim et al. (2003) proposes hybrid scheduling and control system architecture for warehouse management. Żak & Jacyna-Gołda (2013) apply a queuing theory to organization, but these and other approaches stay in a different group of methods than presented in the paper.

The issue of internal transport system organization is in practice simplified to descriptive form (see Frazelle, 2002; Fijałkowski, 2012 or Pyza, 2011) or is detailed for specific cases (Ambroziak & Lewczuk, 2008a; Jacyna & Klodawski, 2011 or Bard, Morton & Wang, 2007). Literature review revealed a wide range of scheduling problems in different applications (compare handbooks like Bruckner, 2007) with exact and heuristic solutions. Ambroziak & Lewczuk (2008b) indicate that organization of internal transport system can be classified as scheduling problem, so appropriate solution methods like genetic programming could be used.

Genetic programming in application for optimization problems is popular as can be seen through handbooks like Goldberg (1998) and Michalewicz (1999). The ITO in practice requires only approximate fulfilment of boundary conditions and obtaining the rational solution instead of the optimal one. These make the genetic programming especially useful. Since the genetic algorithms don't work schematically, there is always a possibility of getting rational solutions better than gained intuitively. The basic advantage of these algorithms is ability to search efficiently the solution space and find feasible solutions meeting all constraints or at least those significant.

The basic difficulties with using genetic algorithms to solve scheduling tasks are related to construction of chromosome, selecting inheritance, selection, crossover and mutation rules and mechanisms of fixing damaged individuals, determining additional procedures directing evolutionary process (on the base of detailed information about solved task) and building an initial population.

Finding a feasible solution of ITO means that it is possible to realize the process with disposed resources, while obtaining more than one solution and selecting the better one gives opportunity to realize a process with additional savings. The article contains a concept of genetic algorithm solving ITO task and appropriate discussion.

## 2. Internal transport organization

### 2.1. Input data

The article bases on a mathematical model of internal transport process presented by Ambroziak & Lewczuk (2008b). The selected input data of the problem are defined as follows.

The tasks of internal transport process are numbered by  $i$ ,  $i \in I$ . Each task is described by daily work-load  $L_i^D$ ,  $\mathbf{L} = [L_i^D : i \in I]$  given as a product of number of daily repetitions of transport cycle  $\lambda_i^D$  and its duration  $t_i$ . The work-load can be dependent on used resources  $L_i^D(u, c)$ .

Tasks are performed by transport devices of different types numbered by  $u$ ,  $u \in U$  and employees of different categories  $c$ ,  $c \in C$ . The set  $U = \{0, 1, \dots, U\}$ , where  $u = 0$  means that task is done manually – no device is necessary.  $\text{Card}(U) = U + 1$ . The set  $C = \{0, 1, \dots, C\}$ , where  $c = 0$  means that transport is done automatically – no employee is necessary.

$\text{Card}(C) = C + 1$ . The permissible assignment of resources to the tasks is defined by array  $\mathbf{V} = [v^{i,u,c} \in \{0, 1\} : i \in I, u \in U, c \in C]_{I \times U \times C}$ . If  $v^{i,u,c} = 1$ , then  $i$ -th task can be done by  $u$ -type device with employee of  $c$ -category. The unit costs of labour and devices operation  $k^c$  and  $k^u$  as well as specific time utilization coefficients  $\omega^u$  and  $\omega^c$  are known.

Daily working time of a logistics facility is divided into  $T$  equal time intervals. Each time interval has known duration  $\tau$  (typical value  $\tau = 60$  minutes). The set of time intervals  $T = \{1, t, \dots, T\}$  defines the fragmentation of the schedule.

### 2.2. Technical conditions of internal transport organization

The organization of internal transport process is understood as assigning to each task a number of time intervals constituting a time slots for realization and assigning appropriate resources in line with rules:

- Daily workloads of tasks and capable resources of realization are known.
- The duration of transport cycles as well as routes and technical aspects of conveying are not considered.
- The sequence of tasks and technical requirements of organization are specified by system of constraints.
- All tasks must be finished within time daily operating time  $\tau \cdot T$  (hours or minutes).
- The work intensity for all tasks is constant during a work shifts.
- The tasks can be finished in stages (more than one time slot for a task) until the schedule fragmentation is acceptable.
- The time to change workplace by devices and employees is neglected while it is compensated by time utilization coefficients.

Particular devices (and employees) are not assigned to the tasks but the groups of devices (of the same type) and employees (of the same category) are allocated to tasks realization.

### 2.3. Optimization task of internal transport organization

The binary decision variable was formulated to reflect assignment of time intervals and resources to the tasks. The variable  $x_i^{t,u,c} = 1$  when  $i$ -th task is

to be realized in  $t$ -th time interval by  $u$ -type device and employee of  $c$ -category, and 0 otherwise. Decision variables are described by a matrix  $\mathbf{X} = [x_i^{t,u,c}]_{I \times T \times U \times C}$ .

Different criteria functions can be formulated, but the most comprehensive and conclusive is a function evaluating the daily distribution of work-load for particular groups of devices and categories of employees normalized by operation costs – formula 1, where all markings are described in a text. (compare Fijałkowski, 2003 with Lewczuk, 2011). The construction and interdependency of ITO constrains make difficult obtaining feasible solutions. Constrains reflect typical technological operating conditions in logistics facility and include:

Detailed forms of above constrains are provided by Lewczuk (2011).

Listed constrains in most cases are introduced naturally and can't be overcome (like limited number of devices) but some of them regulate time slots of tasks realization and other organizational elements used for current re-engineering of the process. For that reason constrains can be divided into two groups:

1. Primary – necessary to keep because of hard technological reasons.
2. Secondary – which, under specific conditions, are not obligatory – the process will not stop if these constrains are not abided but the efficiency parameters will change.

The decision about including constrains into one of the groups must be undertaken in relation to particular case and technological requirements.

### 3. Solving the optimization task

#### 3.1. Exact solving

Classic scheduling problems are classified as NP-hard (solving time increases exponentially with the increase of the task – Sysło, Deo & Kowalik, 1995; T'kindt & Billaut, 2006) and there are no efficient algorithms finding optimal solution in polynomial time.

Proposed non-linear optimization task of ITO is a sub-class of scheduling problems (compare with Błażewicz et al., 2007 and Bruckner, 2007). Decisive version of presented problem is NP-complete, so the task is NP-hard. This feature of a problem, together with composite set of constrains and large sizes of practical cases make difficult using exact methods for its rational solution. Then it is necessary to develop approximation algorithms to generate feasible solutions meeting practical requirements, but without the guarantee of their optimality (Bruckner, 2007; Findeisen, Szymanowski & Wierzbicki, 1973).

There are no studies proving explicitly NP-complexity of the issue, but it was estimated on a base of dimensions of potential space of solutions reduced by selected constrains.

$$F(\mathbf{X}) = \max_{t \in T} \left\{ \sum_{i \in I} \left( \frac{L_i^D \sum_{u \in U} \sum_{c \in C} x_i^{t,u,c} (k^u + k^c)}{\tau \cdot \min_{v \in V} \{v^{i,u,c} \cdot (k^u + k^c) : v^{i,u,c} = 1\}} \cdot \sum_{s \in T^i} \sum_{u \in U} \sum_{c \in C} x_i^{s,u,c} \right) \right\} \rightarrow \min_{[\mathbf{X}]} \quad (1)$$

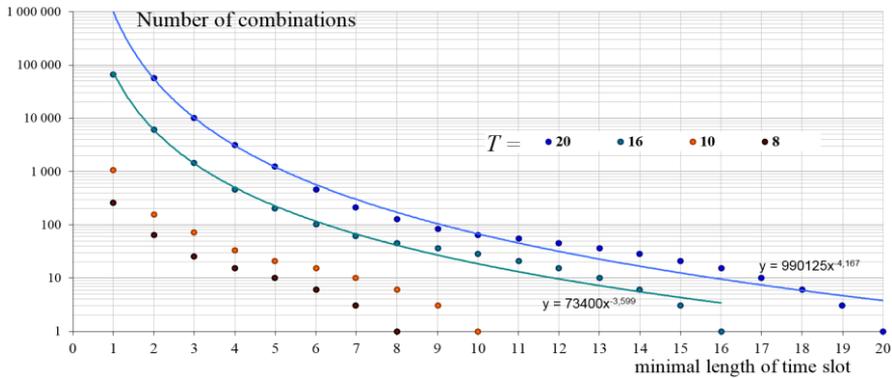


Fig. 1. Number of possible combinations of schedule for one task

The time domain of discussed optimization problem is quantified in order to limit the number of potential solutions (set  $T$ ). Other domains relate to tasks and resources (equipment and employees) and are also discrete (sets  $I$ ,  $U$  and  $C$ ), so the number of solutions is finite. Figure 1 presents the results of brute-force search of possible versions of schedule for one task only and disregarding allocation of resources to the task. The brute-force search was carried for time domain containing  $T = 8, 10, 16$ , and  $20$  time intervals constituting a schedule. Results for  $T = 24$  and  $48$  were approximated. Different minimal lengths of time slots (1, 8, 10, 16 and 20) were investigated for each number of time intervals  $T$ .

Increasing the cardinality of the set  $I$  of tasks will result in an exponential increase in the number of possible solutions, wherein constrains reduces the size of the problem only polynomially. Increasing the number of time periods  $T$  and eliminating the limitation for minimum length of time slot additionally increases exponentially power base.

Type of proposed optimization task, construction of decision variables and the multitude of constrains with a global impact on the quality of the solution (a small number of constrains with local impact on the value of the criteria function) makes difficult using exact methods of solutions and forces using tailored heuristics.

## 3.2. Encoding elements of the ITO optimization task

### 3.2.1. Genetic programming

Genetic programming allows formulating and solving optimization and adaptive tasks with evolutionary mechanisms. The idea of genetic

algorithms assumes reducing decision problem into a chromosome (genome of genes) – a structured set of parameters (genes) describing a single solution. In the next stage chromosomes are evaluated by fitness function and form populations of individuals. Populations are generated by crossing individuals, mutation and selection to improve the subsequent populations containing better individuals – chromosomes with highest evaluation value (Michalewicz, 1999).

### 3.2.2. Determined allotment of time slots and resources to the tasks

Allocation of resources and time slots to the tasks is done by setting binary value of decision variable. The domain of time is represented by  $T$  time intervals of  $\tau$  length. Zero-one time allocation enforces zero-one pursuance (or not) of constrains. Failure to comply with at least one of constrains rejects the solution from the set of feasible solutions. However, some constrains of internal transport process may have less potential importance than others, and their failure to comply under certain circumstances may be more favourable for the process than forcing it primary shape (eg. circumscribing time windows for supplies to strict hours reduces the flexibility and restrict possible system's response for variable material flows).

### 3.2.3. Fuzzy allotment of time slots and resources to the tasks

The allocation of time slots to tasks may by fuzzy – similar to the belonging function in a fuzzy logic. If the fact of task realization in a selected time interval is described by the value of the specified range and

this value describes the power of allocation, it is possible to speak about the fuzzy allocation. It is assumed that 0 is the minimum value and means the total absence of the task in a given time interval, and  $N$  means certainty of task realization.

The decision variable in that case can be written in the following form:  $x_i^{t,u,c} \in (0; N)$ , where  $N > 0$ , is interpreted as the strength of allotment of  $t$ -th time interval to time slot in which  $i$ -th task will be realized. If  $x_i^{t,u,c} > 0$ , then  $i$ -th task will be performed by  $u$ -type device and  $c$ -category employee in  $t$ -th time interval with a probability of  $x_i^{t,u,c} / N$ . The assignment of device and employee to the task retains binary and is not graded.

Such wording of decision variable significantly broadens the scope of implementation genetic programming tools and brings them closer to reality and – at the same time – increases the searching space for the algorithm. Effective implementation of this approach requires determining acceptable levels of allocation strength  $n$ ,  $0 < n < N$ . An additional issue is to determine the possible penalties for failure to comply with constrains. Those penalties should reflect the strength of the allocation  $n$  and the derogation from constrain.

### 3.2.4. Fuzzy allotment of time slots and resources to the tasks

Internal transport process is subjected to disruptions resulting from randomness in supply chain enforcing current reorganization of a process. In addition, the process constraints can be flexible in certain cases and consequences of transgressing may be graded according to the scale of cutting. This approach allows obtaining a greater number of feasible solutions.

Separate but comparable penalties should be determined for each criterion and, if necessary, for each task. Penalties should reflect the scale of cutting, which in case of different constrains will refer to the different technical characteristics of the process and must be included in the objective function.

### 3.2.5. Fuzzy allotment of time slots and resources to the tasks

This representation describes reality the best, but significantly impedes the formal notation and the implementation of task.

## 3.3. The chromosome

### 3.3.1. Chromosome with integer genes for determined allotment of time slots and resources to the tasks

Chromosome representing parameters of a single solution should reflect the decision variables in the manner empowering evolutionary mechanisms. It was decided to discuss a chromosome with integer and binary genes and fuzzy allocation of time slots to tasks.

Decision variables are defined in four dimensions: time, tasks, equipment and employees, then a chromosome consists of three first level sections describing the dimensions of the decision variables array:

- Section mapping the **allocation of employees** of appropriate category to tasks.
- Section mapping **allocation of equipment** of different types to tasks.
- Section mapping **allocation of time intervals** to tasks.

Each first level section is divided into second level sections. The number of second level sections corresponds to the number  $I$  of tasks in the process. The number of genes in each second level section is equal to  $T$  (grain of a schedule). Accordingly, the first level section contains  $I \cdot T$  genes, and the entire chromosome will consist of  $3 \cdot I \cdot T$  integer genes. Chromosome structure is shown in Figure 2.

The range of integers is limited by the cardinality of sets  $U$  and  $C$ . In the first level section mapping the **allocation of employees**, all genes will be therefore integers of  $\langle 0; C \rangle$ , in the section mapping **allocation of equipment** for tasks, genes will be integers of  $\langle 0; U \rangle$ . In the sections mapping **allocation of time intervals** genes will be binary numbers. Under this assumptions it is possible to estimate the number of combinations of chromosome  $K = (2(C + 1)(U + 1))^{I \cdot T}$ . For a typical case:  $C = 2$ ,  $U = 3$ ,  $T = 8$ ,  $I = 7$  the number of combinations will be  $1,95808E + 77$ .

### 3.3.2. Chromosome with binary genes for determined allotment of time slots and resources to the tasks

Binary notation increases the number of genes in the chromosome and can simplify pseudo-evolutionary mechanisms. The structure of chromosome of that construction is shown in Figures 3 and 4.

The estimated number of combinations of chromosome will be  $K = 2^{(C+U+1) \cdot I \cdot T}$ . For a typical case:  $C = 2, U = 3, T = 8, I = 7$  number of combinations will be 1,3998E + 101.

The construction of a chromosome also bases on two-level sections, but the number of first level sections is  $C+U+1$ . Each first level section corresponds to the binary allotment of employees and equipment for tasks. The structure and number of second level sections will not change. Each first level section consists of  $I \cdot T$  genes, and the entire chromosome is composed of  $(C+U+1) \cdot I \cdot T$  of a binary genes.

3.3.3. Chromosome with integer genes for fuzzy allotment of time slots and resources to the tasks

In case discussed in point 3.2.3 the allocation of time intervals for tasks is fuzzy. Sections structure is the

same as presented in 3.3.1. Therefore, the first level section contains  $I \cdot T$  genes, and the entire chromosome will consist of  $3 \cdot I \cdot T$  integer genes. The structure of chromosome of that construction is shown in Figure 5.

The variation range of integers is limited by the cardinals of sets  $U$  and  $C$  and range  $N$ . The first level section mapping the assignment of employees to tasks contains integer genes with values in range  $\langle 0; C \rangle$ . Section mapping assignment of equipment for tasks contains genes in range  $\langle 0; U \rangle$ . Section mapping decision about carrying out tasks in  $t$ -th time intervals is composed of integer genes in range  $\langle 0; N \rangle$ . Under this assumptions it is possible to estimate the number of combinations of chromosome  $K = ((C+1)(U+1)(N+1))^{I \cdot T}$ . For a typical case  $C = 2, U = 3, T = 8, I = 7, N = 9$  number of combinations will be 2,7174E + 116.

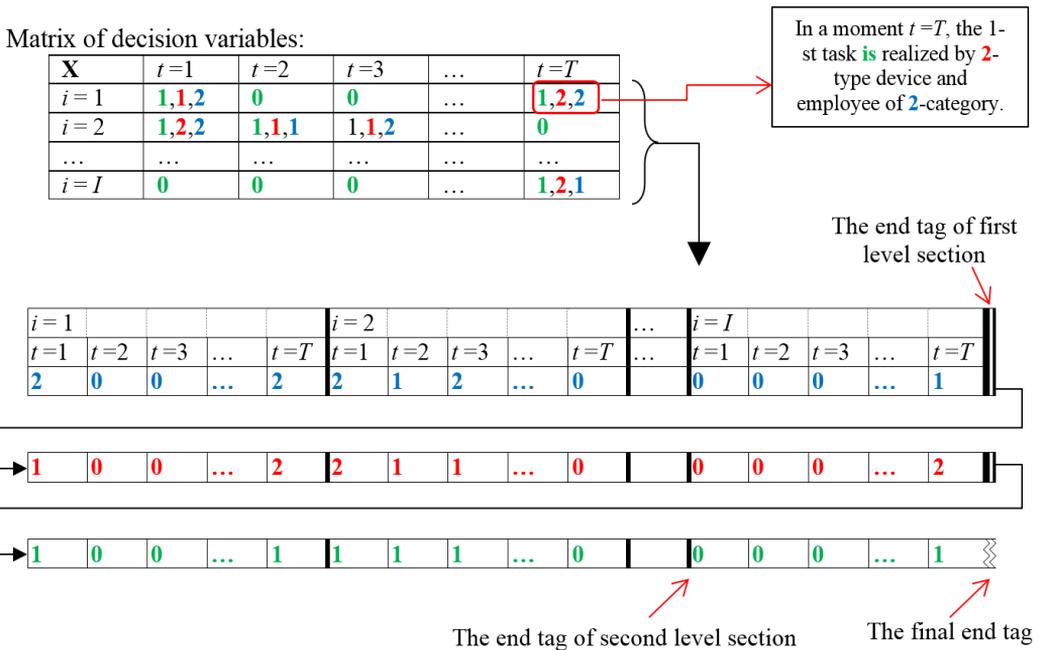


Fig. 2. Graphical representation of a chromosome with integer genes for determined allotment of time slots and resources to the tasks

For  $C=\{0,1,2\}$  and  $U=\{0,1,2\}$ ,

Decision about time allotment

<b>X</b>	$t=1$	$t=2$	$t=3$	...	$t=T$
$i=1$	1	0	0	...	1
$i=2$	1	1	1	...	0
...	...	...	...	...	...
$i=I$	0	0	0	...	1

Variables for  $c=1$

<b>X</b>	$t=1$	$t=2$	$t=3$	...	$t=T$
$i=1$	0	0	0	...	0
$i=2$	0	1	0	...	0
...	...	...	...	...	...
$i=I$	0	0	0	...	1

Variables for  $c=2$

<b>X</b>	$t=1$	$t=2$	$t=3$	...	$t=T$
$i=1$	1	0	0	...	1
$i=2$	1	0	1	...	0
...	...	...	...	...	...
$i=I$	0	0	0	...	0

Variables for  $u=1$

<b>X</b>	$t=1$	$t=2$	$t=3$	...	$t=T$
$i=1$	1	0	0	...	0
$i=2$	0	1	1	...	0
...	...	...	...	...	...
$i=I$	0	0	0	...	0

Variables for  $u=2$

<b>X</b>	$t=1$	$t=2$	$t=3$	...	$t=T$
$i=1$	0	0	0	...	1
$i=2$	1	0	0	...	0
...	...	...	...	...	...
$i=I$	0	0	0	...	1

In a moment  $t=T$ , the 1-st task is realized by 2-type device and employee of 2-category.

Fig. 4

Fig. 3. Graphical representation of a chromosome with binary genes for determined allotment of time slots and resources to the tasks – array of decision variables

Fig. 3

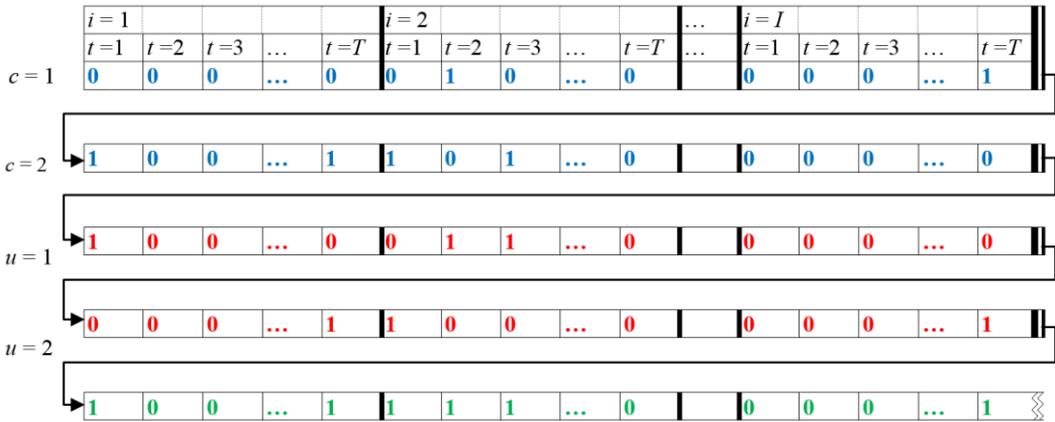


Fig. 4. Graphical representation of a chromosome with binary genes for determined allotment of time slots and resources to the tasks – chromosome structure.

The array of decision variables can be:

<b>X</b>	$t=1$	$t=2$	$t=3$	...	$t=T$
$i=1$	9,1,2	0	0	...	8,2,2
$i=2$	2,2,2	5,1,1	8,1,2	...	0
...	...	...	...	...	...
$i=I$	0	0	0	...	4,2,1

In a moment  $t=T$ , the 1-st task is realized by 2-type device and employee of 2-category.

The end tag of first level section

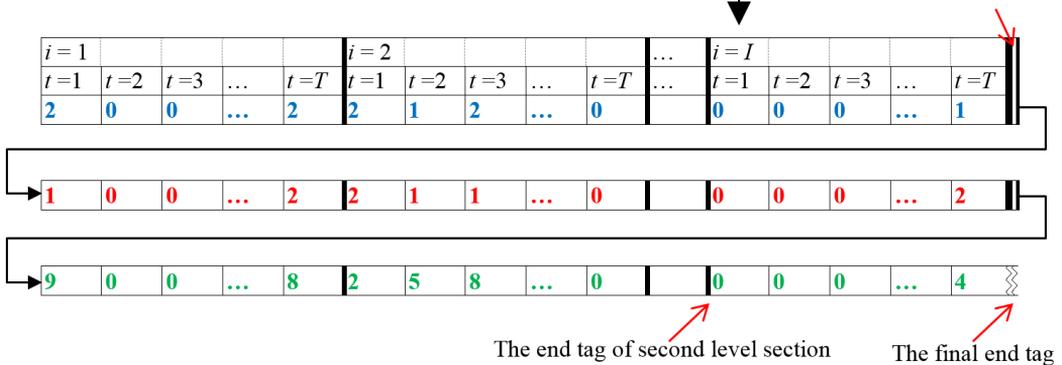


Fig. 5. Graphical representation of a chromosome with integer genes for fuzzy allotment of time slots and resources to the tasks

## 4. The algorithm of evolutionary program

### 4.1. Iterative loop of evolutionary program

The program solves ITO optimization task through implementing an iterative loop of the following form:

1. Generating initial population.
2. If the creation of the initial population is not possible, go to step 10.
3. Selection of individuals to crossover.
4. Crossing and repair of "damaged" individuals – construction of a descendants group.
5. Selection of descendants and parents – construction of a new population.
6. Mutation and possible repair of "damaged" individuals.
7. Archiving individuals best suited to the environment.
8. Archiving average value of quality for the population.
9. If the end condition has not been met, then return to step 3.
10. End – selection of the best individual archived in step 7.

### 4.2. Initial population

Generating feasible solutions to combine an initial population for NP-hard problems with complicated structure of constrains limits the generation of the initial population and is a complex (or NP-hard) problem (Homenda, 2008). Accordingly obtaining the initial population may be carried out in several basic ways:

- a) The initial population consists solely of subjects meeting a full set of constrains.
- b) The initial population includes individuals not complying with secondary constrains.
- c) The initial population includes individuals not complying with primary and/or secondary constrains.

When generating the initial population, one should seek to maximize the number of individuals meeting complete set of constrains. In complex systems of constrains (the use of all proposed constrains tangled with each other), the greatest difficulty in generating initial solutions is related to allocation of time to tasks, as shown by experiments. The allocation of equipment and workers is less problem. For this reason, it is necessary to develop an algorithm to generate initial solutions, which primarily assigns time intervals for tasks and resources in next step.

Due to the fact that the initial population is expected to be diverse, each individual should be formed separately from the other (preferably randomly).

The size of initial population must be appropriate. If it contains too few individuals, the algorithm can stop in one of the local minima. On the other hand, too large population reduces the efficiency of the algorithm. In this case the binary representation has an advantage wherein regardless of the type of problem the newly created individual contains a specific number of randomly generated bits. The following algorithm for generating initial solutions can be used:

1. Assign constrains as primary or secondary.
2. Reduce the time domain by excluding sections constrained for use by primary constraints.
3. (optional) Reduce the time domain by excluding sections constrained for use by secondary constraints.
4. Generate initial population by random assigning time intervals to tasks within the limits as described in step 2.
5. (optional) Generate initial population by random assigning time intervals to tasks within the limits as described in step 3.
6. Random assignment of resources to tasks (under constrains for number of devices and employees) for individuals generated in step 4.
7. (optional) Random assignment of resources to tasks (under constrains for number of devices and employees) for individuals generated in step 5.
8. (optional) Including additional random individuals to the initial population without giving primary and / or secondary constrains.

Depending on the chromosome (as in sections 3.3.1 to 3.3.3), draws are conducted within appropriate limits set by cardinals of data sets  $C$ ,  $U$  or scale  $N$ . An alternative algorithm for generating the initial population involves the following steps:

- 1 - 3 as previous.
4. Manual construction of a single feasible solution.
5. Random change of selected genes allocating particular time intervals to tasks and random redistribution of equipment and employees to make new solutions.

An alternative approach would be more likely to give individuals keeping a full set of constrains or at least primary ones, but can potentially lower the number of good quality individuals.

### 4.3. Fitness function and the selection environment

The criteria function given in point 2.3 will evaluate the fitness of individuals to the environment. For chromosomes encoded like presented in point 3.3.1 and 3.3.2 (determined and unequivocal allocation of time and resources to tasks) finding value of fitness function is not a problem. In addition the chromosome constructed in that manner allows using other criteria functions as mentioned in point 2.3.

Difficulties arise when the value of the criterion function is given by fuzzy assignment (as in point 3.3.3). Fuzzy time slots are better solution for natural selection mechanisms used by genetic algorithms. It enlarges space of solutions affecting the probability of falling into local extreme of fitness function. The fact of not complying with primary and secondary constraints and the "strength" of the overrun must be then included into fitness function. For this purpose the *reducing rates* are formulated to consider the potential impact of breaking constrains on a final solution.

Assessment of solutions quality in terms of failure to comply with selected constrains may be performed as follows:

1. Assignment of weights  $m_o \in \langle 0;1 \rangle$  to particular constraints, where  $m$  is the weight,  $o$  is a number of constraint, and 1 means the total inability to skip constraint and 0 otherwise.
2. Describe all constrains as primary and secondary on a base of their weights.
3. If setting a value to decision variable  $x_i^{t,u,c} \in \langle 0;N \rangle$  breaks the  $o$ -th constrain, then the impact of this on the criterion function (depending on whether it is acceptable to brake primary and/or secondary constrains) is determined as follows:

$\forall x_i^{t,u,c} \in \mathbf{X}$  breaking the constrains:

$$\left( 1 + \sum_{o=1}^O \frac{x_i^{t,u,c}}{N} m_o \right) \cdot F(\mathbf{X}) \quad (2)$$

where  $F(\mathbf{X})$  is a current value of fitness function for a given array of variables  $\mathbf{X}$ , and  $O$  is a total number of constrains.

Weighting constrains should be done experimentally for a particular case. Effective solving real cases will

be possible under the following rules (stemming from experiments):

- Fragmentation of schedule should be rough; the longest possible time slots should be allotted to the tasks.
  - The universal equipment and employees should be used in order to not limit the algorithm
  - The number of constraints and the scope of their application should be minimal.
  - Make as many constrains assigned to the secondary.
  - Avoid constrains sequencing tasks which significantly impede solving the ITO task.
- If possible, the types and categories of labour should be permanently assigned for tasks.

### 4.4. Genetic operators

#### 4.4.1. Selection

The selection of individuals in a population can be done through:

- roulette – if the entire population is reflected as the circle then each individual has assigned a pie proportional to its goodness, which increases its chances of being drawn to crossover operation.
- Ranking / linear ranking – individuals are ranked to indicate the order of selection of individuals to be crossed, the method gives a quick solution clearly superior to random solutions, but is conducive to falling into the local extremes,
- tournament – tournament groups are drawn from the population. A dominant individual is selected for multiplications from each group. The method makes algorithm's convergence to the global optimum slower, but provides a greater diversity of genes and reduces the likelihood of falling into a local extremum,
- mixed – connecting selected features of the above. In case of ITO with the chromosome set out in point 3.3.1, the selection will be done by roulette method with additional random individuals with poor match introduced to the population to diversify the pool of genes.

The equal size of pool is adopted – it consists of the same number of individuals in each generation. This solution allows for control of progress of average match of the population, and thus to observe a genetic algorithm operation. It also can result in possible limitation of evolutionary mechanisms and increases a chance to fall into local extreme of

adjustment function, however, is better due to time estimation and control on calculations.

The selection of new population occurs through a combination of:

- A specific number of children created in crossover  $\delta^{ch}$  (roulette method).
- A specific number of parents  $\delta^{pr}$  (roulette method)
- A specific number of parents and children with value of fitness function below the average for the population  $\delta^{avg}$  (draw).

Not selected individuals are killed, but before that all individuals are evaluated due to the fitness function. Population size is then  $\beta = \delta^{ch} + \delta^{pr} + \delta^{avg}$ . The number of individuals of each category must be set individually and experimentally for considered case. The roulette method can be replaced by ranking or tournament.

If the crossover process rises less than  $\delta^{ch}$  feasible individuals, then the difference is covered from a pool of parents. In case of tasks with a complex system of constraints resulting in too small initial population in relation to predetermined value  $\beta$ , killing any individuals, should be forbidden until population reaches the size of  $\beta$  individuals.

#### 4.4.2. Crossover

Crossover is a genetic operator used to vary the programming of a chromosome from one generation to the next. Crossover is a process of taking more than one parent solutions and producing a child solution from them by exchanging batches of genes. There are methods for selection chromosomes, but not all individuals must take a part in. Crossover is determined by certain factors.

If the individual has been chosen to be crossed, it takes crossbreeding in "everyone with everyone". A couple of individuals give two descendants.

Crossover should reflect the natural mechanisms of evolution and generate "healthy" individuals with a high probability. This means a solution that can be evaluated by fitness function, and may participate in the next crossing. Such an assumption requires strict rules.

A multipoint crossing mechanism will be used (as discussed by Goldberg, 1998), with random intersections cutting chromosomes of pair of parents. Two or three intersection points are drawn to exchange respectively one or two sections of genes. Due to the structure of the chromosome (division into sections of 1st and 2nd level) the PMX, CX or OX methods will not be used.

#### 4.4.3. Crossover – repair of individuals

The exchange of genes likely causes failure to meet some ITO constraints. In this case, depending on the version of the problem, it is possible:

- Optional killing of descendants not meeting primary and / or secondary constraints.
- Attempting to repair them.

Repair of individuals requires reviewing each decision variable in matrix  $\mathbf{X}$  and verifying breaking any constraint (even if penalties are introduced). For each type of constraint a possible repair method should be determined. The effective repair of genes can be implemented only in relatively simple cases like transgressing time window, or assigning a device that can't perform the task.

#### 4.4.4. Mutation

New population established by crossover and selection must be subjected to mutations. The following principles are applied:

- Any gene in population can be mutated.
- The mutation of the gene is determined by probability (relatively low).
- Mutation is to replace the information stored in the gene by other random information from specified range of variation given by technical conditions.
- If the mutating information stays in accepted range of variation, but causes the disobedience of primary and / or secondary constraints, it is possible to:
  - make further attempts, until it succeeds or exceeds the limit of repetitions.
  - abandon attempts to mutate a particular gene.
  - if the secondary constraint is broken, and it is acceptable – don't take any actions.
  - attempt to repair the gene (point 4.4.3).
- The coefficient of mutation probability change  $\gamma$  is applied to increase the probability of mutations in each iterative step of algorithm. Such coefficient prevents the homogeneous population in terms of genotype. With a fixed number of generations it should be activated at the end of the algorithm operation. With unspecified number of generations it can be time-activated, or linear change of mutation is possible. Mutation coefficient should be determined experimentally.

#### 4.4.5. Characteristics of the population

It was pre-established that launching genetic algorithm solving ITO task, needs specifying:

- The cardinality of the population. It is assumed that the population has a fixed size of  $\beta$ . The basic difficulty is generating initial population which (under certain systems of constrains) is almost impossible. For this reason, in cases of relatively simple structure of constrains the size of the population will be from 40 to 80 individuals. In case of the complex structure of constrains the maximum number of individuals that were able to generate should be adopted as the population size.
- The number of generations. It needs to be determined experimentally by observing the convergence of fitness function. The number of generations is dependent on the run of evolutionary process, as well as the time spent on the operation of the algorithm. Number of generations of 500 is an initial value. With no limit on the number of generations, the convergence chart should be observed in real-time.
- Parameters  $\delta^{ch}$ ,  $\delta^{pr}$  and  $\delta^{avg}$ , are selected on the basis of an experiment. The initial values for these parameters are:  $\delta^{ch} = 0.5\beta$ ,  $\delta^{pr} = 0.3\beta$  and  $\delta^{avg} = 0.2\beta$ .
- The crossing probability should be relatively high. The initial value for each of the individuals is 0.8.
- The mutation probability should be relatively low. The initial value for each gene is 0.04.
- The coefficient of mutation probability change  $\gamma$  must be set experimentally depending on the number of generations and whether it is fixed. The impact of the coefficient  $\gamma$  should be observed in the convergence chart. The initial value for the number of generations 500 is 1.2. This means that the probability of mutation in the last generation will be the 120% of the initial value in the first generation. Of course, the probability of mutation must not be greater than 1.
- The number of cuts in the parental chromosomes is dependent on the decision: 2 or 3. Only disingenuous sections of chromosomes should be exchanged between parents.

**5. Preliminary experiments**

A computer tool was prepared in order to carry out computational experiments. The tool allows to represent internal transport process along with its basic characteristics, equipment and labour categories to perform manual experimenting with its organization (Figure 6). It allows tracking meeting or not constrains, current tracking of fitness function value according to changes in the schedule or insight into the current performance and cost parameters as a function of the schedule.

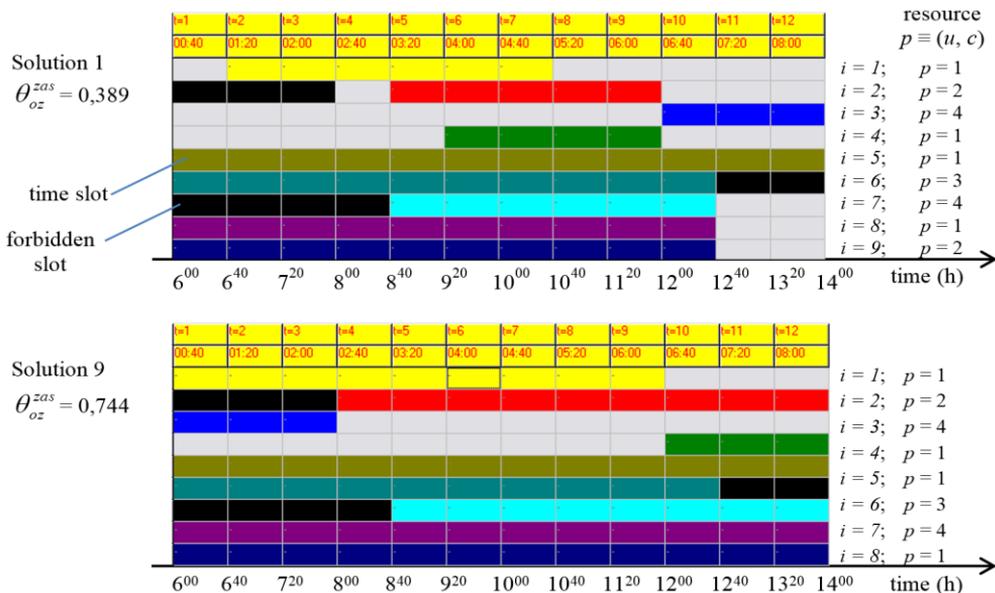


Fig. 6. Schedules of internal transport with assigned resources

The tests with the simple form of genetic algorithm described above were carried out. The tests led to the following conclusions:

- Attempts were done for the chromosome with determined allotment of time to tasks (binary notation), and fixed allocation of resources. The fixed allocation eliminated the difficulties associated with some constrains limiting the availability of equipment and personnel (like maximum number of devices embodying the task).
- Obtaining initial population meeting the complex system of restrictions, especially for cases with the number of tasks of more than 5 and the number of time intervals greater than 8 is a difficult issue. In such cases achieved results were significantly worse than the intuitive solution of the problem. The size of the initial population was not sufficient for the effective operation of the algorithm.
- Reducing system of ITO constrains significantly improves the situation. Fitness function in such a case quite rapidly converges to a steady value (a chart for a simple case shown in Figure 7). Distinct changes in the matching visible at the end of the operation are a result of changes in the probability of mutation.

The genetic programming seems to be a proper tool for ITO problem. Presented discussion about conceptual usage of genetic algorithm is a base for further research

## 6. Conclusions

Efficiency and productivity of logistics facilities depend on many factors resulting from supply chain construction, characteristics of clients and handled materials, structure of material flows, architectural solutions, applied handling and information technologies and organizational issues. The last of these factors can be in some cases easily improved within a certain range by simple actions. A proper organization of internal transport by smart allocation of time, handling equipment and workers to the task is one of these actions. Simple intuitive solutions of internal transport organization are in most cases sufficient, but possibility of better solutions to be found forces developing new organization methods. Proposed optimization task of ITO gives an opportunity to increase the efficiency and to lower operational costs, but is too complex to be solved effectively for real cases. That makes genetic programming preferred to solve the task and give applicable results. Presented genetic algorithm with its components was constructed with regard to crucial features of the ITO and takes into account typical sets of data available in most logistics facilities. A proper application of the algorithm will be a base to solve the task and gain useful results not only in designing stage, but also in everyday operation practice.

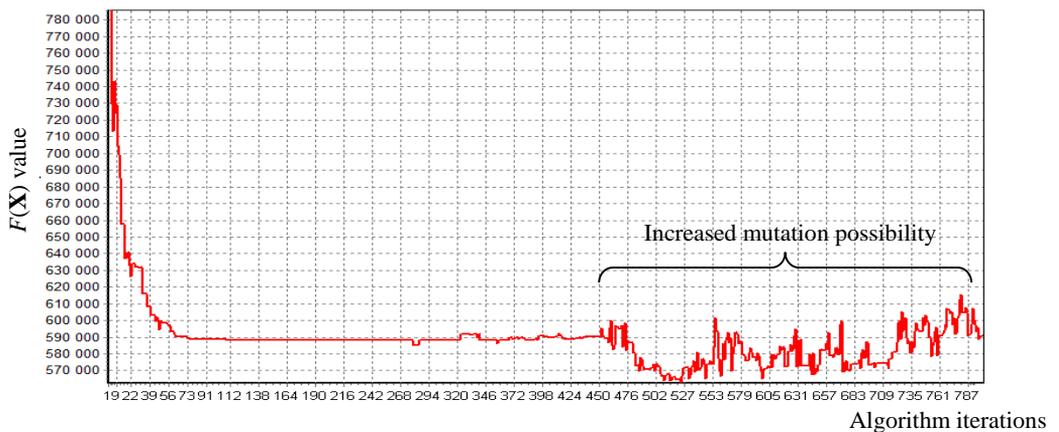


Fig. 7. Sample graph of convergence for trivial case of ITO

**References**

- [1] AMBROZIAK, T. and LEWCZUK, K., 2008a. A method for scheduling the goods receiving process in warehouse facilities. *Total Logistic Management*, 1, pp. 7-14.
- [2] AMBROZIAK, T. and LEWCZUK, K., 2008b. Wybrane aspekty harmonogramowania procesu magazynowego. *Prace Naukowe PW Transport*, 64, pp. 5-12.
- [3] AMBROZIAK, T., 2007. Metody i narzędzia harmonogramowania w transporcie. 1 edn. Warsaw: Wydawnictwo Instytutu Technologii Eksploatacji.
- [4] BARD, J.F., MORTON, D.P. and WANG, Y.M., 2007. Workforce planning at USPS mail processing and distribution centers using stochastic optimization. *Annals of Operations Research*, 155(1), pp. 51-78.
- [5] BŁĄŻEWICZ, J., ECKER, K.H., PESCH, E., SCHMIDT, G. and WEGLARZ, J., 2007. Handbook on scheduling: from theory to applications. Springer-Verlag Berlin Heidelberg.
- [6] BRUCKER, P., 2007. Scheduling algorithms. 5th edn. Springer-Verlag Berlin Heidelberg.
- [7] FIJAŁKOWSKI, J., 2003. Transport wewnętrzny w systemach logistycznych: wybrane zagadnienia. Warsaw: Oficyna Wydawnicza Politechniki Warszawskiej.
- [8] FIJAŁKOWSKI, J., 2012. Racjonalizacja potencjału magazynowego w systemach logistycznych – procedury analityczne i projektowe z przykładem. *Logistyka*, 2012(4), pp. 141-152.
- [9] FINDEISEN, W., SZYMANOWSKI, J. and WIERZBICKI, A.P., 1973. Metody obliczeniowe optymalizacji. 1st edn. Warsaw: Wydawnictwa Politechniki Warszawskiej.
- [10] FRAZELLE, E., 2002. World-class warehousing and material handling. 1st edn. New York: McGraw-Hill.
- [11] GOLDBERG, D.E., 1998. Algorytmy genetyczne i ich zastosowania. Warsaw: Wydawnictwa Naukowo-Techniczne.
- [12] HOMENDA, W., 2008. Algorytmy, złożoność obliczeniowa, granice obliczalności. Warsaw: Centrum Studiów Zaawansowanych Politechniki Warszawskiej.
- [13] JACYNA, M. and KŁODAWSKI, M., 2011. Matematyczny model kształtowania strefy komisjonowania. *Automatyka*, 15(2), pp. 183-193.
- [14] KIM, B., HERAGU, S.S., GRAVES, R.J. and ONGE, A.S., 2003. A hybrid scheduling and control system architecture for warehouse management. *Robotics and Automation, IEEE Transactions on*, 19(6), pp. 991-1001.
- [15] KŁODAWSKI, M. and JACYNA, M., 2011. Czas procesu kompletacji jako kryterium kształtowania strefy komisjonowania. *Logistyka*, 2011(2), pp. 307-317.
- [16] LEWCZUK, K., 2011. Organizacja procesu magazynowego a efektywność wykorzystania zasobów pracy. *Logistyka*, 2011(4), pp. 563-570.
- [17] MICHALEWICZ, Z., 1999. Algorytmy genetyczne + struktury danych = programy ewolucyjne. 2nd edn. Warsaw: Wydawnictwa Naukowo-Techniczne.
- [18] PYZA, D., 2011. Multi-criteria evaluation of transportation systems in supply chains. *Archives of Transport*, 23(1), pp. 47-65.
- [19] SYSŁO, M.M., DEO, N. and KOWALIK, J.S., 1995. Algorytmy optymalizacji dyskretnej. Warsaw: Wydawnictwo Naukowe PWN.
- [20] T'KINDT, V. and BILLAUT, J., 2006. Multicriteria scheduling: theory, models and algorithms. Springer-Verlag Berlin Heidelberg.
- [21] ŻAK, J. and JACYNA-GOŁDA, I., 2013. Using Queue Theory to Analysis and Evaluation of the Logistics Centre Workload. *Archives of Transport*, 25-26(1-2), pp. 117-135.