Application of Genetic Algorithms in Design of Public Transport Network

Piotr Lesiak, Piotr Bojarczak
Radom University of Technology, Poland

The paper presents possibilities of application of genetic algorithms in design of public transport network. Transportation tasks such as determination of optimal routes and timetable for means of transport belong to difficult complex optimization problems, therefore they cannot be solved using traditional search algorithms. It turns out that genetic algorithms can be very useful to solve these transportation problem.

Keywords: genetic algorithms, search methods, optimization, transportation problems.

1. INTRODUCTION

Genetic algorithms are a relatively new branch of science developed at the University of Michigan by John Holland in 20th century. They try to imitate processes of natural selection occurring during species breeding [6], [10], [14]. In the nature, the strongest (the most fitted) individuals are likely to survive and their genes will participate in the creation of offspring. This cycle is continuously repeated and each new generation (offspring) obtained after each pass is better fitted for survival in its environment than its parents. This feature can be utilized to solve difficult complex N-P complete optimization problems for which differential cannot be determined analytically. Therefore genetic algorithms are mostly used in optimization problems for which traditional methods fail.

Many transportation tasks can be regarded as complex optimization problems, for instance, determination of optimal routes and timetables for means of transport, control of the traffic flow using traffic signal lights, dynamic coast control of train movement and so on. In last decade, genetic algorithms are more and more often applied to solve sophisticated transportation problems. This paper presents issues which are relevant to genetic algorithms and their application in chosen transportation problems. It consists of four sections. The first section is the introduction and the last one presents conclusions. Second section describes the idea of genetic algorithms and relevant terminology. The third presents the design of the mass transit route network using genetic algorithms.

2. THE IDEA OF GENETIC ALGORITHMS

Genetic algorithms are search algorithms which are based on concepts of natural selection and natural genetics [10]. These algorithms determine the number of potential solutions called population and encode each potential solution to specific problem on chromosome-like data. The genetic algorithm differs from other search methods in that it searches among a population of points, and works with a coding of parameters set, rather than the parameter values themselves. While searching, the algorithm uses objective function information without any gradient information. The transition scheme of the genetic algorithm is probabilistic, whereas traditional methods use a gradient information and are deterministic [6].

The working principle of genetic algorithms can be presented in fig. 1 and described by the following pseudo-code:

Formulate initial population
Randomly initiate population
repeat
  evaluate objective function
  find fitness function
  apply genetic operators

selection (reproduction)
crossover
mutation
until *stopping criteria*

Each of the item of this pseudo-code will be discussed briefly in following subsections.

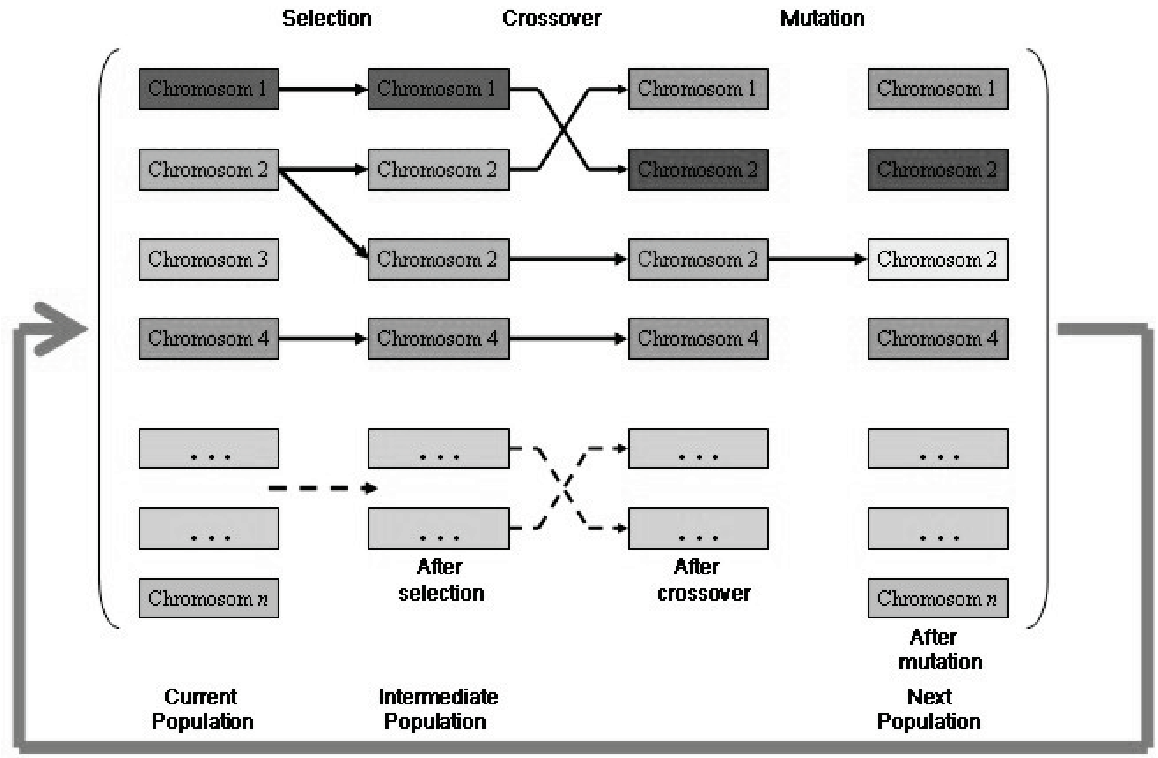

Fig. 1. Illustration of the working principle of the genetic algorithm.

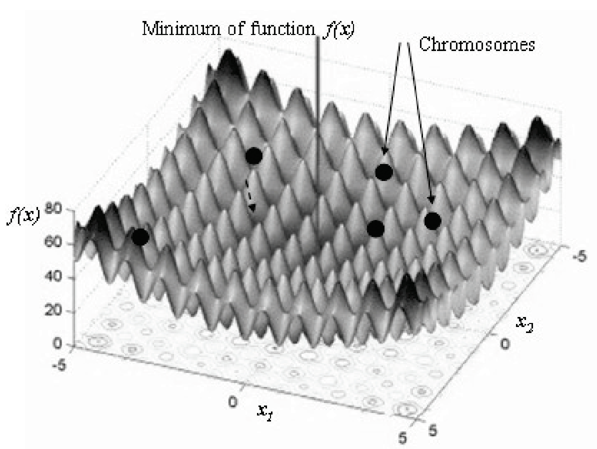

Fig 2. Exemplary distribution of the initial population consisting of 5 chromosomes.

2.1. THE CODING MANNER AND THE CHOICE OF POPULATION SIZE

At the beginning of the algorithm it is necessary to determine the coding manner which allows for the coding of variables describing the problem. The most common coding method is to transform the variables to binary string or vectors. If the problem has more than one variable, a multi-variable coding is constructed by concatenating as many single variables as the number of variables in the problem. The number of coding bits per a single variable depends on the accuracy of expected solution. Let *LB*, *HB* denote minimal and maximal values of the coding variable respectively and *n* is the number of coding bits per the variable, then the accuracy *Ac* of the coding can be calculated as:

$$Ac = \frac{HB - LB}{n - 1} \tag{1}$$

The other crucial issue which has a significant impact on the performance of the algorithm is the choice of the number of chromosomes (solutions) included in the population. There exists a trade-off between the time spent by the algorithm for finding the solution and its precision. The larger number of chromosomes increases the precision of the solution and slows down the performance of the algorithm and vice versa. The number of chromosomes included in the populations depends on the complexity of the problem and usually varies between 10 and 50. At the beginning of the algorithm, initial values for chromosomes (population) are usually set randomly. Fig. 2. shows the exemplary distribution of initial population which consists of 5 chromosomes.

a)

b)

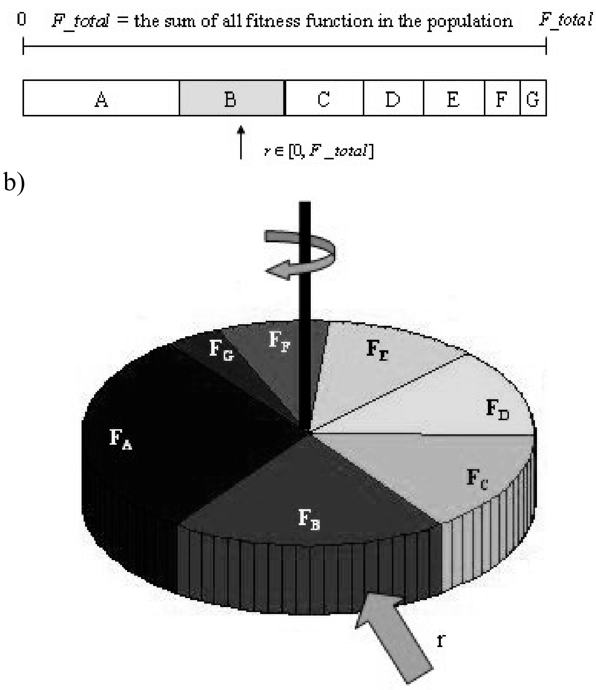

Fig. 3. Illustration of roulette selection process.

2.2. *FITNESS FUNCTION AND SELECTION PROCESS*

Genetic algorithms (GA) mimic the survival of the fittest principle of nature to make a search process. Therefore, GA are naturally suitable for solving optimization problems [10]. A fitness function *F(***x***)* is derived from the objective function (function which undergoes optimization) and used in successive genetic operations. Fitness is a quality value which is a measure of the reproductive efficiency of chromosomes (individuals). Fitness is used to allocate reproductive traits to the chromosomes in the population and thus act as some measure of goodness to be optimized. This means that individuals with higher fitness value will have higher probability of being selected as candidates for further examination. There are two forms of fitness function. The first using in minimization process is of the following form:

$$F(\mathbf{x}) = \frac{1}{f(\mathbf{x}) + 1} \tag{2}$$

The second using in maximization process can be formulated as:

$$F(\mathbf{x}) = f(\mathbf{x}) \tag{3}$$

where $f(\mathbf{x})$ denotes the objective function and $\mathbf{x} = [x_1, x_2, \cdots, x_m]$ is a vector of variables being optimized.

Selection (reproduction) is an operator that makes more copies of better chromosomes (having high value of fitness function) in an intermediate population. Then crossover and mutation are applied to the intermediate population to create the next population. The process of going from the current population to the next population constitutes one generation in the execution of a genetic algorithm. There are several schemes for selection process: roulette wheel selection and its extensions, scaling techniques, tournament, and ranking methods [10].

The most common and the simplest is the roulette wheel selection method. In this case, all fitness functions F_A, F_B, ... F_G ,in the population are arranged in descending order (fig. 3a) and their sum *F_total* is calculated. Next, the variable $r \in < 0; F\_total >$ is selected at random, and the chromosome is copied to intermediate population when the variable *r* occurs inside the interval defined by fitness function corresponding to this chromosome. This cycle is repeated N times. After N cycles, the number of copies of each chromosome occurring in the intermediate population is proportional to the fitness function. This method is similar to spinning the roulette wheel (fig 3.b).

2.3. *CROSSOVER OPERATION*

A crossover operator recombines two chromosomes from an intermediate population to crossing site as shown in Fig. 4. The portion right of the selected site of these two chromosomes are exchanged to form a new pair of the chromosomes. The new chromosomes are thus a combination of the old chromosomes.

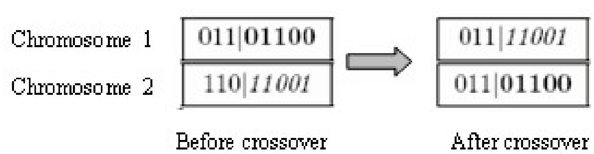

Fig. 4. Illustration of a one site crossover.

Two site crossover is a variation of the one site crossover, except that two crossover sites are chosen and the bits between the sites are exchanged. The underlying objective of the crossover is to exchange information between chromosomes to get chromosomes that are possibly better than their parents.

2.4. MUTATION OPERATION

Mutation adds new information in a random way to the genetic search process and ultimately helps to avoid getting trapped at local optima [10]. It is an operator which introduces diversity in the population whenever the population tends to become homogenous due to repeated use of selection and crossover operations. Mutation may cause the child chromosomes to be different from their parent chromosomes. The need for mutation is to create a point in the neighbourhood of the current point, thereby achieving a local search around the current solution. In mutation, each bit in each chromosome obtained after crossover operation is flipped with some low probability p_m usually less than 0.01. Such low probability value ensures that only very few bits or even none of bits in each chromosome may be changed, what in turn allows for a local search around the current solution.

2.5. STOPPING CRITERIA

After the process of selection, crossover and mutation is complete, the next population can be evaluated. This cycle constitutes one generation. This processes are repeated until some stopping criteria is not met. There exist two kind of stopping create two better chromosomes. In this process, two chromosomes are picked from intermediate population at random and some portion of chromosomes are exchanged between them. There are two types of crossover operation: a one site crossover and two site crossover [14]. A one site crossover operation is performed by randomly choosing a crossing site along the chromosome and by exchanging all bits on the right side of the criteria [10]. In first case the operation of the algorithm is terminated when the fixed number of generations is reached. Then the chromosome with the highest fitness function is the solution of the problem. In the second case the operation of the algorithm is terminated when fitness functions for all chromosomes in the population converge to some value. It means that further generations do not improve the fitness functions and all chromosomes have the same values. Then the value of any chromosome (individual) in population constitutes the solution of the optimization problem. In practice, due to faster termination the first stopping criteria is usually applied.

3. GA IN THE DESIGN OF THE TRANSIT ROUTE NETWORK

The design of the transit route network comprises transit routing and scheduling problem. In transit routing, a route is to be determined on which transit units (for example buses or subway trains) will run as per some-defined schedule. The purpose of transit routing is to determine a good sets of routes. A good sets of routes should satisfy the following criteria [3], [4]:

- the route set should satisfy most if not all, of the existing transit demand (the requirement of the people to travel);
- the route set should satisfy most of the demand without requiring passengers to transfer from one route to another;
- the route should offer low travel time (including the time spent by passengers in transferring) to its passengers.

In transit routing, the geometry of the path is sought which should be optimal or near optimal from some perspective what implies that it is difficult (NP hard) combinatorial optimization problem. Due to such difficulties, this problem cannot be solved using traditional optimization technique [1], [12]. Nevertheless, there exist some heuristic algorithms which try to tackle transit routing problems [2], [11]. Another optimization problem related to the design of the transit route network is the scheduling of transit units (buses or trains). This task consists in the determination of optimal timetable (frequencies) for each given route. It is realized by minimization of the waiting time of passengers while operating within a set of resource and service related constraints. The resource and service related constraints mean [5], [8], [13]:

- limited fleet size: only fixed number of transit units are available for operating on the different routes;
- limited transit unit capacity: each transit unit has a finite capacity;
- policy headway: on a given route a minimum frequency level needs to be maintained;

– maximum transfer time: no passenger should have to wait too long for a transfer.

The design of transit route network using genetic algorithm is presented on the example of Heavy-Capacity Metro in Taipei city [7]. The metro consists of 54 stations and 24 candidate set routes.

Before defining the objective function, the total time spent by passenger during its travel must be given. This time can be divided into following passenger actions (links) (fig. 5) [7]:

- enter the origin station and walk to the platform (a);
- wait on the platform and board the train (b);
- in the train (c);
- alighting (d);
- walk out on the platform and exit the destination station (e).

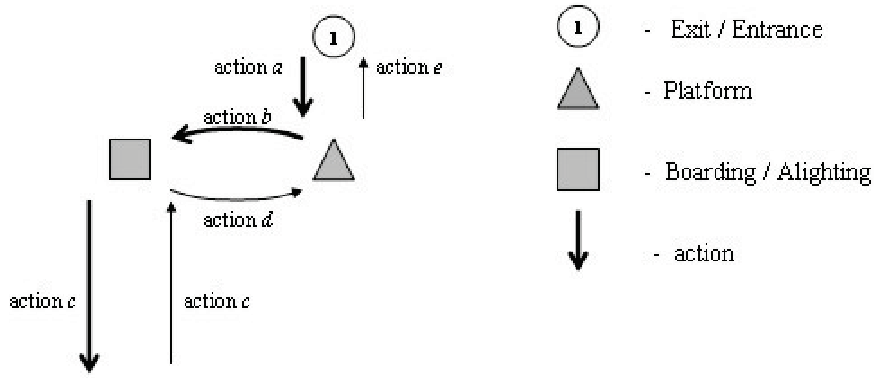

Fig. 5. Passenger actions during its travel.

The objective function consists of two main components expressed in monetary cost: the first representing the generalized cost of transit system and the second representing the cost being incurred by passengers. The first component corresponds to both the energy cost and the depreciation cost. The energy cost is the product of total trip distance of all routes and conversion parameter α. The depreciation cost is the product of total trip time of all routes and conversion parameter β. The conversion parameters α and β reflect unit operation cost per train-km and depreciation cost per train-hour respectively. The second component corresponds to travel cost which is the product of total time value of all passengers and conversation parameter γ. The conversation parameter γ is individually determined by the designer. The objective function can be presented as [7]:

$$Min \ \ \alpha \times \sum_{i=1}^{n} RL_i \times F_i \times s_i + \beta \times \sum_{i=1}^{n} RT_i \times F_i \times s_i + \times \gamma \times (2 \sum_{a=1}^{n_a} \int_0^{x_a} TC(x)dx + 2 \sum_{b=1}^{n_b} \int_0^{x_b} TC(x)dx + \sum_{c=1}^{n_c} \int_0^{x_c} TC(x)dx + 2 \sum_{d=1}^{n_d} \int_0^{x_d} TC(x)dx + 2 \sum_{e=1}^{n_e} \int_0^{x_e} TC(x)dx) , \qquad (4)$$

TC(x) is the passenger travel cost defined according to the suggestion of the U.S. Bureau of Public Roads as [9]:

$$TC(x) = t_0 \times \left[1 + 0.15 \times \left(\frac{x}{v_c}\right)^4\right], \qquad (5)$$

Where t_0 is free travel cost when the link is not congested; x is a passenger volume on the link; v_c is link capacity whose values correspond to:

– design capacity of walk lane for a-type passenger action (link);
– design capacity of platform for b-type passenger action (link);
– route capacity for c-type passenger action (link);
– design capacity of platform for d-type passenger action (link);
– design capacity of walk lane for e-type passenger action (link).

RT_i and RL_i are round trip time for *i-th* route and round trip distance for *i-th* route respectively. s_i is set to 1 when *i-th* route is chosen, otherwise s_i is 0. Let F_i denote the frequency associated with *i-th* route and TS_i be the number of trains required by *i-th* route, then:

$$TS_i = s_i \times RT_i \times F_i \tag{6}$$

During minimization of the objective function the following constraints are imposed [7]:

$$\sum_{i=1}^{n} \left| RT_i \times F_i \times s_i \right| \le TRC, \tag{7}$$

$$x_a = \sum_k f_k^{rs} \times \delta_{ak}^{rs}, \quad a = 1 \sim n_a,$$

$$x_b = \sum_k f_k^{rs} \times \delta_{bk}^{rs}, \quad b = 1 \sim n_b,$$

$$x_c = \sum_k f_k^{rs} \times \delta_{ck}^{rs}, \quad c = 1 \sim n_c,$$

$$x_d = \sum_k f_k^{rs} \times \delta_{dk}^{rs}, \quad d = 1 \sim n_d,$$

$$x_e = \sum_k f_k^{rs} \times \delta_{ek}^{rs}, \quad e = 1 \sim n_e,$$

$$f_k^{rs} \ge 0,$$

$$x_a, x_b, x_c, x_d, x_e \ge 0, \tag{8}$$

$$\sum_{i=1}^{n} \theta_{ij} \times s_i \ge 1, \ for\ all\ line\ segment\ j, \tag{9}$$

$$\frac{1}{\sum_{i=1}^{n} \theta_{ij} \times F_i \times s_i} \ge OT, \ for\ all\ line\ segment\ j \tag{10}$$

$$\frac{1}{\sum_{i=1}^{n} \theta_{ij} \times F_i \times s_i} \le PT, \ for\ all\ line\ segment\ j. \tag{11}$$

$$F_i \ge 0, \tag{12}$$

Where: θ_{ij} is binary variable that is 1 if segment j is served by the ith route, otherwise is 0. Segment means a road between two subsequent stations. *TRC* denotes fleet size in trains. *OT* and *PT* are minimal and maximal allowable headway of segments (in hours) respectively. $\delta_{ak}^{rz}, \delta_{bk}^{rz}, \delta_{ck}^{rz}, \delta_{dk}^{rz}, \delta_{ek}^{rz}$ are binary variables that are 1 if links of type *a, b, c, d, e* are included in kth route for rzth origin-destination, 0 otherwise. n_a, n_b, n_c, n_d, n_e denote number of links of type *a, b, c, d, e*. f_k^{rz} represents the trip on the kth path of alternative paths from rth station to the zth station.

The length of chromosome (solution) equals the number of candidate routes n. Each gene in chromosome corresponds to the number of trains used by the route associated with this gene. A route is neglected if the value of associated gene is equal to 0. The frequency for *ith* route (gene) can be calculated readily by:

$$F_i = \frac{TS_i}{RT_i} \tag{13}$$

Fig.6 presents example of chromosome. The first route uses 10 trains. The second route uses 3 trains. The third route uses 0 trains, it means that this route is excluded from the set of routes which are taken into consideration during design. It should be also pointed out that the sum of gene values must not exceed the fleet size. In order to ensure the minimization of the objective function (4), the fitness function presented by formula (2) has been used.

Route	1	2	3	......	$n-1$	n
Chromosome	10	3	0	......	4	2

Fig. 6. Example of chromosome used to code the solution

Initial solutions are generated randomly and should satisfy constraints (5)-(10).

One site crossover has been used in the construction of genetic algorithm. Selection process based on stochastic universal sampling (extension of roulette wheel method) was chosen.

The algorithm is terminated after the fixed number of generations. As it was presented in [7], this genetic algorithm was able to find the optimal solution.

4. CONCLUSIONS

Genetic algorithms can be applied in solution of optimization problems for which traditional methods fail. Transportation tasks, in particular the design of transit route network belong to such group of problems. Because of their combinatorial character (NP hard), they cannot be solved by traditional gradient methods. It turns out that genetic algorithm is a tool which can be very useful in solving them.

REFERENCES

[1] Baaj, M.H, Mahmassani H, An AI-based approach for transit route system planning and design, Journal of Advance Transportation (2), pp. 187-210, 1991

[2] Baaj, M.H, Mahmassani H, Trust: a LISP program for the analysis of transit route configurations, Transport Research Record, vol. 1283, pp. 125-135, 1990

[3] 3. Chakroborty P, Optimal routing and scheduling in transportation: using genetic algorithm to solve difficult optimization problems, Indian Institute of Technology, Kanpur, available on a website.

[4] Chakroborty P, Genetic algorithms for optimal urban transit network design, Journal Computer-Aided Civil and Infrastructure Engineering, Vol.19, pp.184-200, 2003

[5] Chakroborty P, Deb K, Subrahmanyam P.S, Optimal scheduling of urban transit systems using genetic algorithms, ASCE Journal of Transportation Engineering, Vol.121, (6), pp. 544-553, 1995

[6] Houck Ch.R, Joines J. A, Kay M. G, A genetic algorithm for function optimization: a matlab implementation, North Carolina State University, Grant number DMI – 9322834, 1996.

[7] Jin-Yuan Wang, Chih-Ming Lin, Mass Transit route network design using genetic algorithm, Journal of the Chinese Institute of Engineers, Vol.33, No.2, pp.301-315, 2010

[8] Kidwai F. A, Marwah B. R,, Deb K, Karim M. R, A genetic algorithm based bus scheduling model for transit network, Proceeding of Eastern Asia Society for Transportation Studies, Vol. 6, 2005

[9] Lee C.K, Hsieh W.J, A bi level programming model for planning high speed rail service, Transportation Planning Journal, Vol.31. No.1, pp. 95-119, 2001

[10] Mitchell M, An introduction to genetic algorithms, MIT Press, 1999

[11] Van Nes R., Hamerslag R, Immerse B, H, Design of public transportation networks, Transportation Research Record, Vol.4, pp.74-82, 1998

[12] Newell. G.F, Some issues relating to optimal design of bus routes, Transportation Science, Vol.13, (1), pp 20-35, 1979

[13] Tormos P., Lova A., Barber F., Ingolotti L., Abrill M., Salido M.A., A genetic algorithm for railway scheduling problems, Studies in Computation Intelligence, Springer, 2008

[14] Whitley D, A genetic algorithm tutorial, Statistics and Computing, Vol. 4, pp. 65-85, 1994

Piotr Lesiak
Radom University of Technology, Poland
plesiak@pr.radom.net