SIMULATION-BASED FITNESS LANDSCAPE ANALYSIS FOR VEHICLE SCHEDULING PROBLEM

Galina Merkuryeva, Yuri Merkuryev, Vitalijs Bolshakovs

Riga Technical University, Department of Modelling and Simulation,
LV-1076, Riga, Kalku 1, Latvia

galina.merkurjeva@rtu.lv (Galina Merkuryeva)

Abstract

Simulation-based analysis of fitness landscapes with application to the vehicle scheduling problem with time windows is discussed in the paper. Methods of analysis of fitness landscapes and measures known in literature are reviewed. The procedure for simulation based analysis of fitness landscape is introduced. The tool that allows automating this analysis is described. The simulation model is developed within AnyLogic 6 simulation software, while Java applications generate landscape path solutions, analyse their fitness values series and implement a genetic algorithms. Experimental study for a vehicle scheduling problem with the time windows is given and demonstrates the main steps of fitness landscape applied to optimization problem.

Keywords: vehicle scheduling, time windows, simulation, optimisation, fitness landscape.

Presenting Author's biography

Galina Merkuryeva is a full professor at Riga Technical University, Department of Modelling and Simulation, Latvia. She has research interests and experiences in discrete-event simulation, simulation metamodelling and optimisation, decision support systems, supply chain simulation and management, and simulation-based training. She is an editor of the Baltic Journal on Sustainability, Technological and Economic Development of Economy (Vilnius, Lithuania).

1 Introduction

Vehicle scheduling problems (VSP) represent a class of freight transportation problems aimed at assigning a set of scheduled trips to a set of vehicles, in such a way, that each trip is associated with one vehicle, and a cost function for all trips is minimised [1, 2]. This problem is often modified with additional constraints like time windows or capacity constraints. A number of methods to solve VSP problems are proposed in literature, e.g. integer programming, combinatorial methods, heuristics. However, they can be applied only for particular problems and cannot be usable in many real-life applications. Also, commercial scheduling software usually based on heuristic optimisation (Logware, Shortec, etc.) cannot satisfy all problem constraints as well as employ new information that become available after a schedule is generated. Nowadays, evolutionary algorithms are often used to solve complex optimisation problems. However, in some cases they may be not enough computationally efficient. The hardness of an optimisation problem and the ability of the evolutionary algorithm to perform an efficient search of the optimal solution can be determined by using fitness landscape analysis [4-7]. The problem is hard to solve with an evolutionary algorithm if its fitness landscape has a large number of structures, which disturb a search of the global solution. The main landscape feature which influences the problem difficulty for an optimisation algorithm is the landscape ruggedness [8]. To measure the degree of ruggedness of the problem search space, several statistical and information measures are introduced.

In practice, VSP can be also complicated by stochastic processes existing in the system, e.g., when a vehicle moving speed is a random variable. In this case, evaluation of fitness for potential solutions can be done through simulation, and simulation-based optimisation could be used to solve the problem. As simulation technology provides a flexible tool to determine the optimality of each solution, simulation-based fitness landscape analysis becomes an important task.

The paper presents simulation-based fitness landscape analysis for the scheduling problem with time windows. The paper is organised as follows. Section 2 defines the problem statement. Section 3 gives basic information about fitness landscape analysis, and the main statistical and information measures of fitness landscape are defined. Section 4 describes building up of the toll for landscape analysis. Section 5 presents results of the experimental study. The last section presents the summary of performed research.

2 Problem statement

A vehicle schedule defines a schedule of goods deliveries by vehicles from a distribution centre (DC) to a net of shops or supermarkets in twenty-four hours [3]. Vehicles or trucks belong to different groups that have various parameters such as capacity, limited velocity and ownership, i.e. own or rented. Distribution routes for vehicles are fixed. For each route, the following parameters are defined: a sequence of shops (route points), an average time interval for vehicle moving between route points, loading and unloading average time and type of goods to be carried on this route. Furthermore, goods could be only delivered to shops in predefined time windows. For each shop, an average demand per product type is defined. Vehicle capacities are limited and known.

The problem is aimed to assign trucks to routes in order to minimise the total idle time for all trucks. Decision variables that assign trucks to trips and define a start time for each trip are introduced in the problem, i.e. *trip x_truck i*, *trip x_time*, where x is a trip number, and i is a truck number. The objective function f is defined as follows:

$$f = \sum_{i=1}^{N} T_{idle}^{i} \rightarrow \min, \qquad (1)$$

where T_{idle}^{i} is the total idle time for truck i, and N is a number of trucks. The truck idle time is defined as a sum of time periods, when a truck is waiting for the next trip in the parking place. The problem operational constraints are:

1) Truck capacity constraints, which define that a truck cannot ship more freight than its capacity;

2) Delivery time constraints, i.e. goods could be delivered to shops only within defined time windows;

3) Gate capacity constraints, i.e. a number of trucks that can be loaded in a warehouse simultaneously cannot exceed number of gates.

Express analysis shows that the problem could have many solutions not feasible within restrictions. This could make a search process computationally non-efficient. To avoid this, the objective function f is modified taking into account an amount of constraints not satisfied by a potential solution:

$$f^{*} = \sum T_{idle} + k_1 T_c + k_2 T_m + k_3 T_0 + k_4 N_{ol} + k_5 N_{ot}, \qquad (2)$$

where f^* is a modified objective function; T_C defines the total duration of overlapping trips for one vehicle; T_m is the total time of window mismatches; T_o and N_{ol} defines the total time and a number of vehicles that over perform working hours; and N_{ot} is a number of vehicles that are overloaded. Here, all indexes for unsatisfied constraints are multiplied by coefficients $k_i > 1$, $i = 1,\ldots,5$ that increase a value of the objective function and make fitness of these solutions worse.

3 Fitness landscape analysis

The main objective of a fitness landscape analysis is to evaluate the difficulty of an optimisation problem and investigate the ability of the evolutionary algorithms to solve the problem. The fitness landscape consists of three main components: the set of genotypes, the fitness function that evaluates the genotypes and genetics operators that define neighbourhood relations between the genotype sets. A landscape can be interpreted by a surface in a search space that defines the fitness for each potential solution. In this case, searching an optimal solution is interpreted as walking on the fitness landscape surface towards the highest hill, with overcoming other hills and valleys [4].

The structure of the fitness landscape influences the performance of an evolutionary algorithm. There are several characteristics associated with the landscape optima that define the structure of fitness landscapes [5]. These are the landscape modality, epistasis, ruggedness, and deceptiveness. The modality evaluates a number of optima in a search space and an optima density. The epistasis refers to genotype fitness dependence on multiple genes interaction. Both high modality and epistasis lead to a more rugged fitness landscape. Such rugged search spaces are harder to search compared to a smoother landscape with low epistasis and modality [6, 7].

A number of different techniques have been developed for fitness landscapes analysis by evaluating their structural characteristics. These techniques can be divided in two distinct groups: statistical analysis and information analysis.

In statistical analysis techniques, several correlation metrics for characterising the landscape structure are proposed. The autocorrelation function is used to measure the landscape ruggedness. In case of a low autocorrelation between two sets of landscape points separated by some distance, these points have dissimilar fitness values, and the landscape is more rugged. Another correlation metric used in practice is the correlation length. It defines a distance beyond which two sets of fitness points becomes uncorrelated [9]. The magnitude of this length indicates the smoothness of the landscape. A shorter correlation length would indicate less smoother and more rugged landscape.

Information analysis is aimed to obtain more information about the structure of fitness landscapes, comparing to statistical analysis techniques. In particular, it allows estimating the diversity of the local optima, modality of landscape and the degree of regularity of random walks [6]. Information analysis is performed based on the sequence of fitness values obtained by a random walk on the landscape. The concept of entropy proposed in classical information theory is used as a basic concept to quantify the ruggedness of a landscape. Four information measures are proposed in literature [6]. The information content and partial information content are two measures of the entropy or amount of fitness change encountered during the walk in the obtained landscape path. The first one indicates the ruggedness of the landscape, and the second one defines the modality of the landscape path. The information stability characterizes the magnitude of the landscape path's optima. The density-basin information analyses the variety of flat and smooth sections on the landscape. Both statistical and information analysis techniques can be used only for a statistically isotropic fitness landscapes [6].

4 Build up of analysis tool

Simulation is a core technology in which the tool for fitness landscape analysis is build up. Here, fitness evaluation of the solutions in the landscape path is made through simulation. The procedure for the simulation-based fitness landscape analysis [3] comprises of the following steps (Fig. 1):

1) Fitness landscape path generation;

2) Fitness evaluation of solutions in the path through simulation; and

3) Analysis of path fitness sequence.

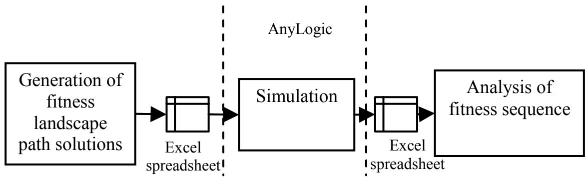

Fig. 1 Fitness landscape analysis stages

To evaluate fitness of solutions in the path, the simulation model in AnyLogic 6 is used. This simulation software is based on the object-oriented conception, implemented in Eclipse framework and employs Java language for the definition of complex structures and algorithms [10]. In the tool, Java applications generate landscape path solutions and analyse their fitness series. As far as AnyLogic software don't have API for external applications, all data is transferred in the form of tables into MS Excel spreadsheets.

At first stage, a random walk on the problem fitness landscape is performed. As a result, genotypes of landscape path solutions are obtained.

At the second stage, the simulation model with different parameters, i.e. solutions with different genotypes, is run. Simulation experiments of the type *'Parameters Variation'* are performed. As a result, the model generates a spreadsheet which contains a sequence of fitness values.

Finally, both statistical and information analysis of the fitness landscape is performed at the same time. The following statistical measures based on the generated path fitness data are calculated: autocorrelation $r(k)$

for sets of points separated with k solutions, and the correlation length τ. The autocorrelation function $r(k)$ is approximated [7] by

$$r(k) \approx \frac{E(f_t f_{t+k}) - E(f_t)E(f_{t+k})}{V(f_t)}, \tag{3}$$

where $E(f_t)$ and $V(f_t)$ represent the expectation and the variance of the sequence of N fitness values $\{f_t\}^N_{t=1}$. The correlation length τ is evaluated [7] by

$$\tau = -\frac{1}{\ln(r(1))}. \tag{4}$$

The tool also provides determination of the following information measures: information stability ε*, information content H(ε), partial information content M(ε) and density-basin information h(ε) for different values of ε. The information content H(ε) is defined [6] by

$$H(\varepsilon) = -\sum_{p \neq q} P_{[pq]} \log_6 P_{[pq]}, \tag{5}$$

where the probabilities $P_{[pq]}$ represents frequencies of possible sub-blocks pq of length two from the string S(ε) with elements $s_i \in \{\bar{1}, 0, 1\}$ enumerated by

$$s_i(\varepsilon) = \begin{cases} \bar{1}, & if \quad f_i - f_{i-1} < -\varepsilon \\ 0, & if \quad |f_i - f_{i-1}| < \varepsilon \\ 1, & if \quad f_i - f_{i-1} > \varepsilon \end{cases}, \tag{6}$$

for any fixed ε. The string S(ε) contains information about the structure of the landscape, and the parameter ε defines the accuracy of calculation of this string.

Partial information content is determined [6, 7] by

$$M(\varepsilon) = \frac{\Phi_s(1,0,0)}{n}, \tag{7}$$

where n is the length of the string S(ε), and the function Φ is calculated recursively by

$$\Phi_S(i,j,k) = \begin{cases} k & , if \quad i > n \\ \Phi_S(i+1, i, k+1), if & j = 0 \ \ and \ \ s_i \neq 0 \\ \Phi_S(i+1, i, k+1), if & j > 0, s_i \neq 0 \ \ and \ \ s_i \neq s_j \\ \Phi_S(i+1, j, k) & , otherwise \end{cases} \tag{8}$$

The information stability ε* is equal to the smallest ε value, when obtained that the fitness path has no structures at all [6]. To define ε*, an interval of possible ε values is divided by 2 at each iteration. A half-interval that contains possible value ε* is selected for further analysis in the next iteration.

Finally, the density-basin information is determined [7] by

$$h(\varepsilon) = -\sum_{p \in \{\bar{1}, 0, 1\}} P_{[pp]} \log_3 P_{[pp]}, \tag{9}$$

where the probabilities $P_{[pp]}$ represents frequencies of sub-blocks pp from the string S(ε).

Measures H(ε), M(ε) and h(ε) are calculated iteratively in the interval [0, ε*) with a step 0.05.

5 Experimental study

5.1 Simulation modelling

The vehicle scheduling model is built as a discrete-event simulation model [11], in which each vehicle is modelled as an active object. The object behaviour is described by a state chart (Fig. 2) that defines vehicle possible states (e.g., parking, loading, moving and unloading) and transition between them. Three classes are defined for shop, trip and job objects. These objects are used for storage of input data of the model. As the amount of input data is large, they are defined in Microsoft Excel spreadsheets and transferred into the model with the help of ODBC (Open DataBase Connectivity). Within model initialisation, it is connected to input data database, and variable collections are set up with data from database.

Parameters of the vehicle schedule (i.e., a trip number and the corresponding truck number) are introduced as control variables in the model and interpreted as decision variables in simulation optimization experiments. The assignment of trucks to trips is transformed to the assignment of jobs into a job list for each vehicle, where a job is defined as a pair of a scheduled trip and start time of this trip. The main performance measure that indicates an efficiency of the vehicle schedule solutions is defined by an average total idle time for all vehicles. In the model screenshot (Fig. 3), the Gantt chart for the vehicle schedule simulated is shown. During simulation, constraint violations such as time window mismatch in delivery (i.e. wrong delivery time), shortage of truck and gate capacity, are fixed.

To see how sensitive the model performance indicators (the total idle time and the total cost of vehicle utilisation) are to input data uncertainty, the following analysis is performed.

The time interval for vehicle moving between route points is defined by a random variable from its mean value with a normal distribution $N(\mu, \sigma^2)$, where $\mu = t_{move}$, $\sigma^2 = k * t_{move}$, where t_{move} is an average value, σ^2 is variance, and the coefficient k which is a ratio of variance to mean characterises the randomness amplitude or data uncertainty. A number of simulation replications n is calculated by a formula [12]:

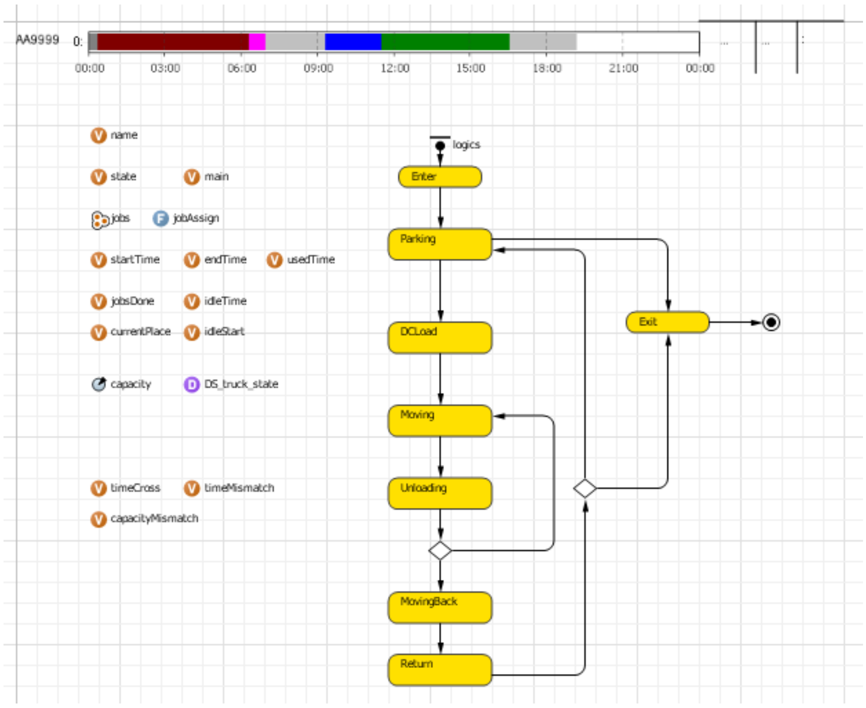

Fig. 2 State Chart

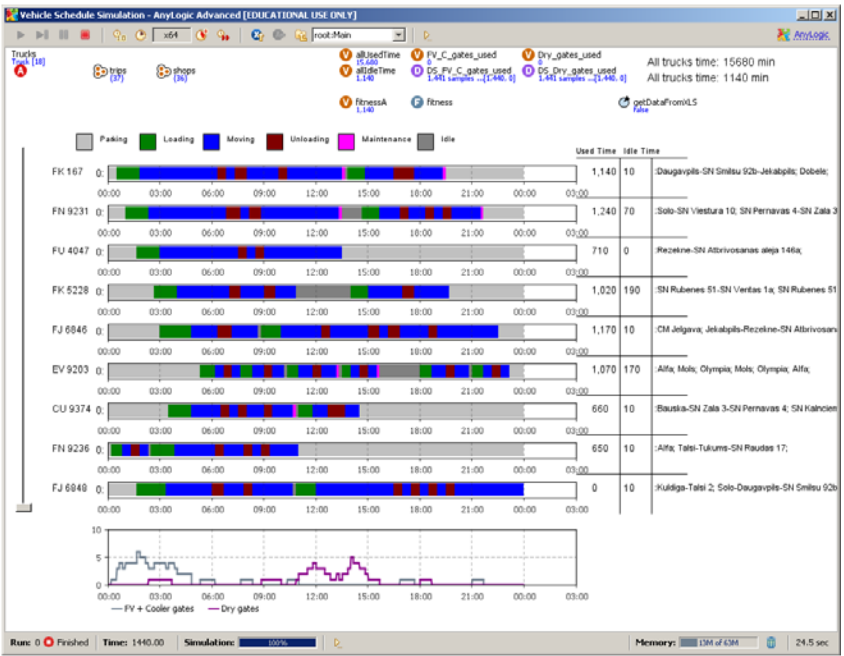

Fig. 3 Gantt chart

$$n = \left[\frac{(Z_{\alpha/2})s}{e} \right]^2, \qquad (10)$$

where e defines an absolute error with significance level $\alpha = 0{,}05$; $Z_{\alpha/2} = t_{\infty,\ \alpha/2}$ and $t_{\infty,\ \alpha/2}$ is Student's coefficient; and s is a sample standard deviation received from experiments. Calculations are made in assumption, that the performance indicator has a normal distribution.

Results of a sensitivity analysis show that the average total idle time of vehicles is dependent on the amplitude of randomness (Fig. 4) while the average total cost of their utilisation is not sensitive to its changes (Fig. 5). Moreover, a growth of uncertainty of moving time between customers leads to the growth of the total idle time. Consequently, this performance indicator is introduced to define the objective function f in (1) and its modification f^* in (2).

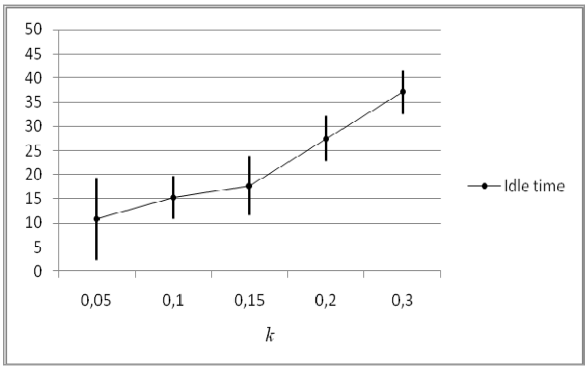

Fig. 4 Dependence of the total idle time on the amplitude of randomness (k)

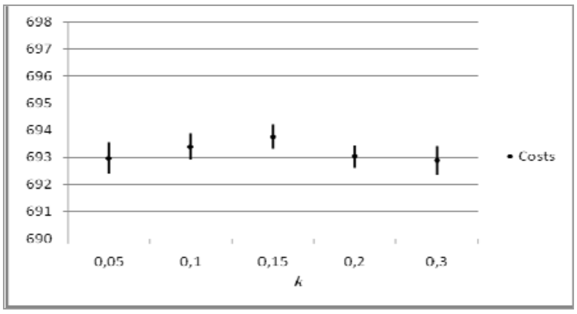

Fig. 5 Dependence of the total cost on the amplitude of randomness (k)

5.2 Landscape analysis

Various series of experiments for fitness landscape analysis with stochastic and deterministic input data were performed. In each series, 5 experiments with landscape 100 solutions' long path were made. In the first series of experiments, the results of fitness landscape analysis for the problem with stochastic input data (stochastic times for moving, unloading, etc.) and deterministic data expressed by average values are compared. Statistical measures are given in Table 1, and informational measures in Table 2.

Tab. 1 Statistical measures

Input data	r(1)	r(10)	τ (0.1)
Stochastic	0.84	0.21	7.24
Deterministic	0.89	0.32	8.75

Tab. 2 Information measures

Input data	H(0.1)	M(0.1)	h(0.1)	ε*
Stochastic	0.66	0.20	0.49	0.40
Deterministic	0.60	0.17	0.37	0.35

Statistical measures indicate that the landscape is relatively rugged and even more rugged for the problem with stochastic data. The autocorrelation between two sets of landscape points separated by 10 solutions is very low. The correlation length is only about 8 solutions. Additionally, the information content H(0.1) is relatively high (more than 0.5). This leads to the same conclusions about the ruggedness of the problem landscape. Then the partial data content M(0.1) is low, and as a result, the modality of fitness landscape is low. Also modality is lower for the problem with deterministic input data. Density-basin information h(0.1) indicates that peaks have high density and their density is higher for a stochastic problem. Information stability value is bigger for a stochastic problem, which means that peaks are higher for stochastic problem.

In the second series of experiments, dependence of informational measures from calculation accuracy ε is analyzed (see, Fig. 6 and Fig. 7). Both information measures decrease when ε is increased.

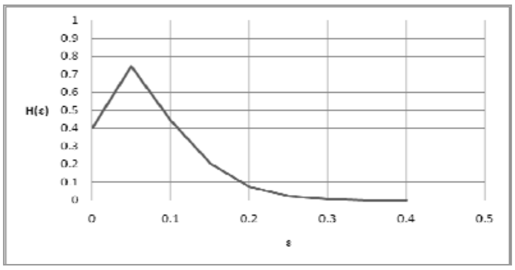

Fig. 6 Information content from accuracy ε

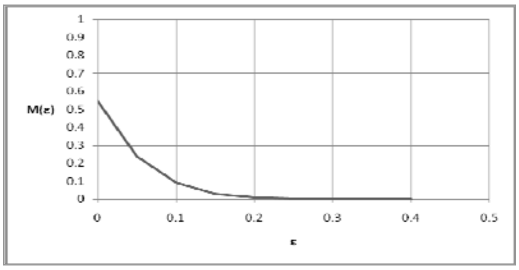

Fig. 7 Partial Information content from accuracy ε

The results of fitness landscape analysis lead to the conclusion that the optimisation problem may be relatively hard for an evolutionary algorithm.

5.3 Simulation optimisation

A genetic algorithm (GA) is applied for search of the best combination of vehicle schedule parameters. It is implemented as Java class and interacts with the simulation model through *Parameter Variation* experiment in AnyLogic. The simulation model is a large dimension model with the number of trips, trucks and shops equal to 37, 17, and 36, correspondingly. The total number of potential solutions is equal to $N = (17 \cdot 134)^{37} = 1{,}69 \cdot 10^{124}$. To simplify the optimisation problem, a schedule for 7 trips is optimised and other trips are fixed. The correspondent number of decision variables is 14, which leads to the number of potential solutions equal to $N = (17 \cdot 134)^{7} = 3{,}18 \cdot 10^{23}$. Function (2) is used for fitness evaluation of the solutions. Loading, moving and unloading times are defined by their average values.

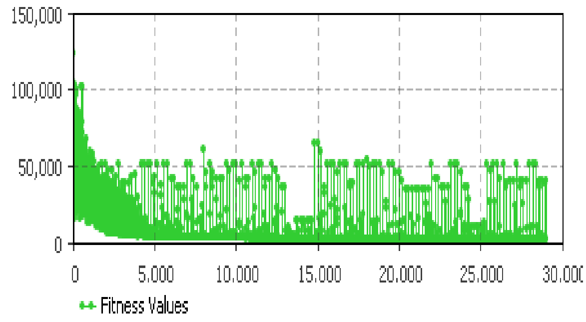

Fig. 8 Convergence of genetic algorithm

Chromosomes are implemented as strings of integer numbers that encode parameters of vehicle schedule, i.e. a vehicle number and start time for each trip. Each population contains 200 chromosomes. All genetic operators are customized for operating with the proposed structure of chromosome. In particular, one point crossover operator for data encoded in real numbers is applied. In the mutation, new random numbers for a vehicle and start time are assigned to one randomly selected trip. Crossover and mutation rates are defined as 70% and 1%, correspondingly, and termination condition is defined by 150 generations. Figure 8 illustrates the performance of GA that converges subject to fitness values to a sub-optimal solution. The solution found allowed decreasing the total idle time comparing with the original schedule.

6 Conclusions

Analysis of the fitness landscape allows evaluating hardness of the optimization problem for evolutionary algorithms. For complex problems fitness landscape analysis require application of simulation. The results of simulation-based fitness landscape analysis show that evolutionary algorithms could provide an efficient tool for solving vehicle scheduling problem with time windows. Future research will be devoted to design of simulation experiments for analysing fitness landscapes and modify a genetic algorithm to allow optimising the problem with a larger number of decision variables.

7 Acknowledgements

This work has been partly supported by the European Social Fund within the project „Support for the implementation of doctoral studies at Riga Technical University".

8 References

[1] D. T. Eiiyi, A. Ornek, S. S. Karakutuk. A vehicle scheduling problem with fixed trips and time limitations. *International Journal of Production* Economics, 117 (1), Jan. 2008, pp. 150-161.

[2] H. Nagamochi, T. Ohnishi. Approximating a vehicle scheduling problem with time windows and handling times. *Theoretical Computer Science*, 393 (1-3), Mar. 2008, pp. 133-146.

[3] G. Merkuryeva, V. Bolshakovs. Simulation-based vehicle scheduling with time windows. *First International Conference on Intelligent Systems, Modelling and Simulation*, Liverpool, 2010. pp. 134-139.

[4] T. Jones. Evolutionary Algorithms, Fitness Landscapes and Search. Albuquerque: The University of New Mexico, 1995.

[5] C. R. Reeves, J. E. Rowe. Genetic Algorithms - Principles and Perspectives. A Guide to GA Theory. Springer, 2002.

[6] V. K. Vassilev, T. C. Fogarty, J. F. Miller. Information Characteristics and the Structure of Landscapes. *Evolutionary Computation*, 8(1), Mar. 2000, pp. 31 – 60.

[7] P. Merz, B. Freisleben. Fitness Landscape Analysis and Memetic Algorithms for the Quadratic Assignment Problem. *IEEE Transactions on Evolutionary Computation*, 4(4), Nov. 2000, pp. 337- 352.

[8] T. Jones, S. Forrest. Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms. *Proceedings of the Sixth International Conference on Genetic Algorithms*, CA, San Francisco: Morgan Kaufmann, 1995, pp. 184-192.

[9] J. T. W. Teo. Pareto Multi-Objective Evolution of Legged Embodied Organisms, Phd thesis. Sydney: University of New South Wales, 2003.

[10]Yu. Karpov. Simulation of systems. Introduction in modeling with AnyLogic 5. BHV Petersburg, 2005. (In Russian)

[11]G. Merkuryeva, V. Bolshakovs. Vehicle Schedule Simulation with AnyLogic. *12th International Conference on Computer Modelling and Simulation*, Cambridge, 2010. pp. 169-174.

[12]Harrell C., Ghosh B. K., Bowden R. O. Simulation using Promodel. McGrawHill, 2004. – 733p.