RIGA TECHNICAL UNIVERSITY

Vitālijs Boļšakovs

SIMULATION-BASED FITNESS LANDSCAPE ANALYSIS AND OPTIMISATION OF COMPLEX SYSTEMS

Summary of Doctoral Thesis

Riga   2013

RIGA TECHNICAL UNIVERSITY
Faculty of Computer Science and Information Technology
Institute of Information Technology

Vitālijs Boļšakovs
Student of the Doctoral study programme „Information Technology”

SIMULATION-BASED FITNESS LANDSCAPE ANALYSIS AND OPTIMISATION OF COMPLEX SYSTEMS

Summary of Doctoral Thesis

Scientific supervisor
Dr.habil.sc.ing., Prof.
G. MERKURYEVA

RTU Press
Riga 2013

UDK 004.94(043.2)
Bo 385 s

Boļšakovs, V. Simulation-based Fitness Landscape Analysis and Optimisation of Complex Systems. Summary of Doctoral Thesis.-R.: RTU Press, 2013.-49.pp.

Printed according to the decision of the RTU Institute of Information Technology Board meeting, June 25, 2013, Protocol No. 13-05.

This work has been supported by the European Social Fund within the project “Support for the implementation of doctoral studies at Riga Technical University”

ISBN 978-9934-10-474-9

DOCTORAL THESIS
IS SUBMITTED FOR THE DOCTOR'S DEGREE
IN ENGINEERING SCIENCE
AT RIGA TECHNICAL UNIVERSITY

The defence of the thesis submitted for the doctoral degree in engineering science (Information Technology) will take place at an open session at the Faculty of Computer Science and Information Technology of Riga Technical University, in 1/3 Meza Street, auditorium 202, at 14^{30}, on December 9, 2013.

OFFICIAL REVIEWERS

Dr.habil.sc.ing, Prof. Zigurds Markovičs
Riga Technical University, Latvia

Dr.sc.ing., Assoc. Prof. Pēteris Grabusts
Rezekne Higher Education Institution, Latvia

Dipl.-Ing. Dr., Prof. (FH), Stefan Wagner
Upper Austria University of Applied Sciences, Austria

DECLARATION

I hereby confirm that I have developed this thesis submitted for the doctoral degree at Riga Technical University. This thesis has not been submitted for the doctoral degree at any other university.

Vitālijs Boļšakovs……………………………(Signature)

Date: ....

The doctoral thesis is written in English. It consists of introduction, 4 sections, conclusions, bibliography and 2 appendixes. It includes 73 figures and 14 tables. The thesis is printed on 135 pages. The full bibliography comprises 94 entries.

GENERAL DESCRIPTION OF THE THESIS

Research motivation

In various cases, traditional optimisation methods (linear programming, integer programming, stochastic optimisation, etc.) could not be applied to solve hard optimisation problems. These methods may lead to ineffective solutions for such problems due to a high number of parameters of an optimised system, existence of stochastic parameters and a large solution search space. A number of metaheuristic optimisation techniques are applied for the optimisation of these tasks. To choose an appropriate technique, fitness landscape analysis of an optimisation problem is performed. Moreover, simulation of the system allows for evaluation of the system performance without analytical calculations. At the present time, simulation optimisation technology is a necessary tool in optimisation of complex systems, where solution evaluation can be complicated (Kleijnen, Merkuryev, Gosavi, Merkuryeva, Carson, Azadivar, etc.). Simulation-based fitness landscape analysis provides an efficient approach to analysis of suitability of the optimisation algorithms.

Nowadays, fitness landscape analysis methods are used for the determination of the problem hardness for the metaheuristic algorithms (Vassilev, Stadler, Affenzeller, etc.). However, there is a lack of information about the application of these methods. Additionally, there is almost no research on the application of simulation in fitness landscape analysis within simulation optimisation of complex systems. Simulation-based fitness landscape analysis will allow better selection of algorithms for optimisation of a complex system, as well as allowing for construction and adjustment of the most appropriate algorithm.

The goal and the tasks of the thesis

The thesis is aimed at developing methods and algorithms for the simulation-based fitness landscape analysis and optimisation of complex systems.

To achieve this aim, the following tasks are specified:

1. To make a review of existing measures and methods of fitness landscape analysis for application in simulation-based optimisation of NP-hard problems.
2. To develop a formalisation scheme for simulation-based optimisation enhanced by a fitness landscape analysis.
3. To perform an analysis of benchmark fitness landscapes in order to define how typical landscape structures influence statistical and information measures of fitness landscapes.
4. To develop a procedure for a simulation-based fitness landscape

analysis and optimisation for NP-hard problems.

5. To perform approbation of the developed methods in optimisation of the delivery planning and scheduling problem.

Theses to be defended

1. Determination of the formal definitions of a fitness landscape and its structures, and a review of the information and statistical fitness landscape analysis methods allows developing a formal scheme for a simulation-based optimisation procedure supplemented with the fitness landscape analysis.
2. By expanding fitness landscape analysis methods into simulation-based optimisation tasks, it is possible to implement algorithms and a tool for application of fitness landscape analysis in the solution of hard optimisation tasks.
3. The experimental analysis of benchmark fitness landscapes allows finding relations between fitness landscape structural properties, landscape measures and behaviour of optimisation algorithms.
4. With application of the developed methods for optimisation of a vehicle route schedule, it is possible to improve delivery planning solutions at the operational level.

The research object and subject

The object of the research is metaheuristic optimisation of NP-hard optimisation problems. The subject of the research is development of methods and algorithms for simulation-based fitness landscape analysis and its application in optimisation of complex systems.

The research methods

The research is based on using system analysis, simulation techniques, simulation-based optimisation, genetic algorithms, statistical and information fitness landscape analysis and metaheuristic optimisation methods.

The scientific novelty

Following scientific novelties are expected in this research:

1. Development of the simulation-based fitness landscape analysis algorithm.
2. Review of an application of a fitness landscape analysis in applied research.
3. Application of the developed methods and algorithms in the delivery planning and scheduling.

Practical value

The developed methods are applied to optimisation of a delivery planning and scheduling problem for a regional distribution centre.

Optimisation problems, which are defined in an integrated approach are analysed with fitness landscape analysis techniques, and conclusions on the problems search space are given.

A fitness landscape analysis and customised optimisation algorithms are applied for optimisation of goods deliveries from a distribution centre to a large retail network of stores. The acknowledgment signed by the chairman of the Board of HAVI Logistics SIA confirms that algorithms and methods developed in the thesis are useful and applicable for large logistic enterprises in Latvia.

Approbation of the obtained results

The results of the thesis have been presented at **12 international scientific conferences**:

1. Riga Technical University 53rd International Scientific Conference Dedicated to the 150th Anniversary and The 1st Congress of World Engineers and RPI/ RTU Alumni, Riga, Latvia, October 11-12, 2012.
2. International conference "*The 24th European Modeling & Simulation Symposium*" (EMSS-2012), Vienna, Austria, 19-21 September 2012.
3. International conference "*25th European Conference on Operational Research EURO-2012*", Vilnius, Lithuania, July, 8-11 2012.
4. International conference "*1st Australian Conference on the Application of Systems Engineering*" (ACASE'12). Sydney, Australia, February 6-8, 2012.
5. International conference "*UKSim 5th European Symposium on Computer Modeling and Simulation*" (EMS2011), Spain, Madrid, November 16-18, 2011.
6. International conference "*13th European Conference on Computer Aided System Theory Eurocast-2011*", Las Palmas de Gran Canaria, Spain, February 6-11, 2011.
7. International conference "*UkSIM Fourth European Modelling Symposium on Computer Modelling and Simulation*" (EMS2010), Pisa, Italy, November 17-19, 2010.
8. Riga Technical University 51st International Scientific Conference, Riga, Latvia, October 13-15 2010.
9. International conference "*The 7th EUROSIM Congress on Modelling and Simulation*". Prague, Czech Republic, September 6-10, 2010.
10. International conference "*12th International Conference on Computer Modelling and Simulation*" (UKSim2010), Cambridge, United Kingdom, March 24-26, 2010.
11. International conference "*1st International Conference on Intelligent Systems, Modelling and Simulation*" (ISMS2010), Liverpool, United Kingdom, January 27-29, 2010.
12. Riga Technical University 50th International Scientific Conference, Riga, Latvia, October 14-16, 2009.

The results have been published in **13 scientific papers**, including **1** book chapter published by Springer and **1** paper in the international scientific journal. The paper "Simulation Optimisation and Monitoring in Tactical and Operational Planning of Deliveries" is awarded by "The 24th European Modeling and Simulation Symposium Best Paper Award". The complete list of publications:

1. Merkuryeva G., Bolshakov V. Simulation Optimisation and Monitoring in Tactical and Operational Planning of Deliveries // Proceedings of the European Modeling and Simulation Symposium, 2012, Austria, Vienna, 19.-21. September, 2012. - pp 226-231. Indexed in: Scopus.
2. Pitzer E., Vonolfen S., Beham A., Affenzeller M., Bolshakov V., Merkuryeva G. Structural Analysis of Vehicle Routing Problems using General Fitness Landscape Analysis and Problem Specific Measures // 1st Australian Conference on the Application of Systems Engineering (ACASE'12), Australia, Sydney, 6.-8. February, 2012. - pp 36-38.
3. Merkuryeva G., Bolshakov V. Simulation-Based Fitness Landscape Analysis and Optimisation for Vehicle Scheduling Problem // EUROCAST 2011, Part I, LNCS 6927: Springer-Verlag Berlin Heidelberg, 2012. - pp 280-286. Indexed in: SpringerLink, Scopus.
4. Merkuryeva G., Bolshakov V., Kornevs M. An Integrated Approach to Product Delivery Planning and Scheduling // Scientific Journal of RTU. 5. series., Computer Science. - 49. vol. (2011), pp 97-103. Indexed in: EBSCO, CSA/ProQuest, VINITI.
5. Bolshakov V., Pitzer E., Affenzeller M. Fitness Landscape Analysis of Simulation Optimisation Problems with HeuristicLab // Proceedings of the UKSim 5th European Symposium on Computer Modeling and Simulation, Spain, Madrid, 16.-18. November, 2011. - pp 107-112. Indexed in: IEEE CS Digital Library, Scopus.
6. Merkuryeva G., Bolshakovs V. Benchmark Fitness Landscape Analysis // International Journal of Simulation Systems, Science and Technology. - Vol.12, No.2. (2011) pp 38-45. Indexed in: Inspec, Scopus.
7. Merkuryeva G., Bolshakov V. Simulation-Based Fitness Landscape Analysis and Optimisation for Vehicle Scheduling Problem // EUROCAST 2011 Computer Aided Systems Theory Extended Abstracts. IUCTC Universidad de Las Palmas de Gran Canaria, Spain, 2011. – pp. 254-255.
8. Merkuryeva G., Bolshakovs V. Structural Analysis of Benchmarking Fitness Landscapes // Scientific Journal of RTU. 5. series., Computer Science. - 44. vol. – Riga: “RTU Publishing House”, 2010. - pp 81-86. Indexed in: EBSCO, CSA/ProQuest, VINITI.
9. Merkuryeva G., Bolshakovs V. Comparative Analysis of Statistical and Information Measures for Benchmark Fitness Landscapes // Proceedings of the UkSIM Fourth European Modelling Symposium on Computer Modelling and Simulation, Italy, Pisa, 2010. - pp 96-101. Indexed in: IEEE CS Digital Library, Scopus.
10. Merkuryeva G., Merkuryev Yu., Bolshakovs V. Simulation-Based Fitness Landscape Analysis for Vehicle Scheduling Problem // Proceedings of the 7th EUROSIM Congress on Modelling and Simulation. Czech Republic, Prague, 6.-10. September, 2010. – p. 88.
11. Merkuryeva G., Bolshakovs V. Vehicle Schedule Simulation with AnyLogic// Proc. of the 12th International Conference on Computer Modelling and Simulation UKsim-2010. Los Alamitos: “IEEE Conference Publication Service”, 2010. – pp. 169-174. Indexed in: IEEE CS Digital Library, Scopus.
12. Merkuryeva G., Bolshakovs V. Simulation-Based Vehicle Scheduling with Time Windows// Proc. of the 1st UKSim/AMSS International conference on Intelligent Systems, Modelling and Simulation. Los Alamitos: “IEEE Conference Publication Service”, 2010. – pp. 134-139. Indexed in: IEEE CS Digital Library, Scopus.

13. Merkuryeva G., Bolshakovs V. Simulation-based analysis of fitness landscape in optimisation // Scientific Journal of RTU. 5. series., Computer Science. - 44. Vol. – Riga: "RTU Publishing House", 2009. – pp. 39-44 Indexed in: EBSCO, CSA/ProQuest, VINITI.

The results of the thesis were used in the following research grants of the Latvian Council of Science:

1. Research grant No. 09.1201 "Simulation-based optimisation using computational intelligence". Project leader: Dr.habil.sc.ing., Prof. J. Merkurjevs (2009);
2. Research grant No. 09.1564 "Simulation and computational intelligence methods for logistics and e-business optimization". Project leader: Dr.habil.sc.ing., Prof. J. Merkurjevs (2010-2012).

The results of the thesis were also applied in the RTU Fundamental and Applied Research Project No. FLPP-2011/6 "Simulation-based cluster analysis and optimisation of vehicle schedules" in cooperation with HAVI Logistics SIA for developing vehicle routing and delivery scheduling algorithms. Project leader: Dr.habil.sc.ing., Prof. G. Merkurjeva (2011).

Approbation of the methods developed in the doctoral thesis was performed within 3 month research mobility within FP7 ICT project No. FP7-248583 "UNITE – UpgradiNg ICT excellence by strengthening cooperation between research Teams in an enlarged Europe" in Upper Austria University of Applied Sciences, Software Engineering Department, Hagenberg, Austria, 2011.

The structure of the thesis

The doctoral thesis consists of introduction, 4 chapters, conclusions, bibliography and 2 appendixes. The thesis contains 135 pages, 73 figures and 14 tables. The bibliography contains 94 entries. The thesis is structured as follows:

Introduction provides motivation of the research, formulates the goal and tasks of the thesis, defines the research object and subject, lists research methods used in the thesis, and describes the scientific novelty of the thesis, its practical value and approbation of the results obtained in the thesis.

Chapter 1 "Literature Review and Problem Statement" describes simulation-based optimisation methods for hard optimisation problems and reviews the concept of a fitness landscape analysis and its purposes. This chapter gives formal definitions and interpretations of a fitness landscape and its structures. Properties of the fitness landscape that influence behaviour of an optimisation algorithm are reviewed. Techniques for statistical and information fitness landscape analysis are discussed. The problem of the simulation-based fitness landscape analysis is defined, and its formalised scheme is proposed and described.

Chapter 2 "Benchmark Fitness Landscape Experimental Analysis" discusses the landscapes of the benchmark problems and shows how the structures of fitness landscapes affect the statistical and information measures of the fitness landscape analysis. Particularly, eight benchmark landscapes are defined, and both statistical and information analysis of these landscapes is made. Additional experiments performed in the thesis figure out the relevance of these landscape measures with performance of optimisation algorithms, and a detailed review of benchmark landscape structures is given.

Chapter 3 "Simulation-based Fitness Landscape Analysis and Optimisation" presents a procedure and algorithms of the simulation-based fitness landscape analysis. A case study on a vehicle scheduling problem is given, and a simulation model for evaluation of scheduling problem solutions is developed. Three scenarios of the problem solution are described. Simulation-based fitness landscape analysis and optimisation of the vehicle scheduling problem is performed by using the developed tool, as well as in the fitted optimisation framework. Recommendations for optimisation of the vehicle scheduling problem enhanced with the fitness landscape analysis are revealed and described.

Chapter 4 "Application in Product Delivery Planning" describes application of metaheuristic optimisation methods in solving of a combined vehicle routing and scheduling task in goods delivery planning. The formal statement of the vehicle routing problem, description of applied methods and optimisation experiments are given in this chapter. The statement of the vehicle scheduling problem is converted into a route scheduling problem statement, which complements the solution of the routing problem. Then scheduling optimisation methods and experiments are described for the routed solutions.

Results and conclusions of the thesis

Bibliography

Appendixes

THE SUMMARY OF THESIS CHAPTERS

1. Literature Review and Problem Statement

Chapter 1 gives an overview of a fitness landscape analysis in metaheuristic optimisation. Simulation-based optimisation methods for complex systems are discussed, and metaheuristic methods are identified as most suitable for the research topic of this thesis. A place for the fitness landscape analysis in such optimisation is defined. Formalisation and structures of a fitness landscape and specific methods for its analysis are given. The chapter ends with a formalised scheme proposed for a simulation-based fitness landscape analysis.

Simulation-based optimisation for NP-hard problems

Modern optimisation problems in logistics and industry are characterized by large dimensions, uncertainty and nonlinearity. Thus they require more powerful methods in stochastic optimisation than traditional ones, such as non-linear-programming methods or classical algorithms in stochastic dynamic programming. In the thesis, parametric optimisation problems, most common in logistics, are considered. Parametric optimisation is performed to find values for a set of parameters which optimise some performance measure, e.g. minimise a cost or maximise a reward [15]. Mathematically, it can be defined as follows: optimise $f(x_1, x_2, ..., x_k)$, subject to some linear or non-linear constraints involving the decision variables x_1, x_2, ..., and x_k, where f denotes a function of the decision variables.

A factor that strongly influences the hardness of the optimisation problem is computational complexity of the problem. Different complexity classes characterize in which way computational time or space is dependent on the size of input data for a problem's solutions. The thesis focuses on the NP-hard problems [42], which are at least as hard as the problems of the NP-complete class, though NP-complete problems are the hardest problems of the NP (nondeterministic computer, polynomial time) class.

Other factors that strongly influence the hardness of the optimisation problem can be the stochastic nature of the optimised system and the hardness of obtaining the analytical form of the objective function. To find solutions of such complex, large-scale, stochastic optimisation problems simulation-based optimisation is applied.

The literature provides a number of numerical optimisation methods that only need the numerical value of the objective function for any solution candidate. These methods form a natural choice in solving applied complex stochastic optimisation problems, where the closed form of the objective function is frequently unknown, but the function itself can be evaluated numerically [15]. Numerical methods also include metaheuristic optimisation methods, which facilitate finding good solutions to large and complex optimisation problems in a reasonable time with the application of different heuristic and stochastic methods. Although metaheuristic methods don't guarantee that the optimal solution to the problem will be found, there is a high interest for the metaheuristic methods in the applied optimisation of real life problems [12]. Heuristics are methods that provide rules for search algorithms to explore good solutions and avoid worse solutions. A large number of metaheuristic optimisation techniques are implemented within a powerful and flexible optimisation framework HeuristicLab [49].

Traditional and numerical optimisation methods used in simulation-optimisation can be divided into the following groups:

- Gradient Based Search methods;
- Stochastic Optimisation;
- Response Surface Methodology;
- Statistical methods;
- Metaheuristic methods.

Gradient Based Search methods are based on the response function gradient. The gradient is estimated to assess the shape of the objective function and to employ deterministic mathematical programming techniques. Stochastic Optimisation contains methods to find a local optimum for an objective function whose values are stochastic and are not known analytically. Statistical simulation optimisation methods use some additional information on the problem and structure of its simulation model. The Response Surface Methodology is based on approximation of the regression models that fit the output variable of a simulation model. Using the regression analysis the regression function that describes dependence of simulation output on input parameters is obtained, which is easier to optimise [5]. These methods require an ability to estimate a gradient of objective function, and most of them are designed for continuous optimisation problems and are hardly applicable for combinatorial problems.

The group of *metaheuristic methods* has a wide concern within this thesis. These methods rely on the evaluation of solution candidates and on heuristic rules, without additional information about the optimised system. This group includes such main methods as the *Genetic Algorithm* (GA) [14, 17], *Evolution Strategy* (ES) [41], Simulated Annealing [24] and Tabu Search [13].

Genetic algorithm and evolution strategy are population-based evolutionary algorithms that are based on the concept of a natural evolution. The GA works on a population of individuals, which represents solution candidates in the form of a string of genes (chromosome), where each gene encodes corresponding parameters of the solution. A sequence of three genetic operators is iteratively applied in the GA: selection, mutation and crossover. The selection operator removes worse individuals from the population and clones good solutions. Individual quality is determined by its fitness, estimated with a fitness or goal function. The crossover operator stochastically mixes parts of chromosomes in randomly selected pairs of individuals to create new solution candidates. The mutation randomly changes the values of genes in a part of randomly selected individuals.

Evolution strategy is similar to GA, but the selection of individuals in a new population is different, and the mutation operator plays a more important role. In the (μ, λ)-ES strategy, an initial population in each iteration contains μ individuals, which are used to produce the offspring population of λ solutions with the evolutionary operators. The offspring population is reduced to μ best individuals, and a new iteration is performed. In the $(\mu+\lambda)$-ES strategy, λ individuals of the offspring population are mixed in one population with μ best solutions from a previous population, and then this population is reduced to μ best individuals.

Simulated annealing and Tabu search are local search metaheuristic methods which rely on the hill-climbing search method. As hill-climbing may get stuck in suboptimal solutions, these metaheuristics introduce additional features to overcome this problem. Simulated annealing allows for stepping backward, but during the algorithm runtime the allowed decline of the fitness value is gradually reduced to zero. The Tabu search uses so called tabu lists, which disallow the search among already visited solution candidates, thus forcing the exploration of new solutions.

In past research metaheuristic methods have shown good results in the solution of combinatorial analytical problems, and they are easily combined with simulation models; thus they are the most suitable for this thesis. The application of the metaheuristic and other numerical methods becomes more important for especially hard optimisation problems such as NP-hard combinatorial optimisation problems [10].

All the complexity factors lead to the time-consuming optimisation, and hard problems often cannot be solved in a reasonable time. For the new uncommon and possibly not investigated optimisation problems, this can lead to complex decision making for the selection and configuration of the optimisation methods. To make the selection and adjustment of an optimisation method more reasonable, a fitness landscape analysis offers methods for the investigation of the problem’s search space and can be applied to analyse the behaviour of metaheuristic methods in the optimisation of a specific problem.

The concept of fitness landscape analysis

The fitness landscape analysis provides methods and techniques for a mathematical analysis of a search space in hard combinatorial and continuous optimisation problems. It can be applied as a support tool to enhance optimisation of complex systems, and it is widely considered in modern literature on artificial intelligence techniques [19, 21, 27, 44, 50]. In general, the fitness landscape is interpreted [20, 39, 45] as a combination of a fitness function of the optimisation problem and the relationships or a distance metric between the solutions in the search space. Hence, the fitness landscape defines the structures of the search space.

The first notion of the fitness landscape is mentioned in [52], where it is proposed as a tool to interpret the behaviour of the biological evolution. Here, the evolution is considered as a walk on the landscape, where its heights define the fitness of the individuals in specific points of the landscape. The process of evolution tends to the highest peaks, where the individuals are most fit to the natural environment. Here, the definition of the fitness landscape shows that at times evolution has to move individuals from one peak to another, higher peak, through the valley with a low fitness, and these landscape structures impact the evolution process.

This concept of the fitness landscape was transferred to the domain of optimisation problems. It was proposed that the structures of the problem's fitness landscape affect the way, in which a search space is examined by a metaheuristic optimisation algorithm. It was assumed that the fitness landscape analysis of the problem would allow getting more information on the problem's properties dependent on a specific optimisation method, which will guide the optimisation process [19, 39, 50].

One of the objectives of the fitness landscape analysis is evaluation of the difficulty of the optimisation problem. The features of fitness landscape that influence the problem difficulty are ruggedness, modality and epistasis of the landscape [21]. Another opinion is that fitness landscape analysis can be applied to obtain a better understanding of the algorithm performance for different instances of the same problem, to configure optimisation algorithm for specific instances [37]. With the fitness landscape analysis it is possible to get measures of the problem's difficulty, and in turn, the recommended configuration of an optimisation algorithm, while gaining deeper understanding of behaviour of the algorithm in the corresponding problem class. The fitness landscape analysis provides information on internal characteristics of the optimisation problem. It is a powerful tool for deep understanding of optimisation problems in classes however can be a weak tool for enhanced optimisation of a stand-alone optimisation problem [37].

The fitness landscape analysis may be used as a tool for dividing instances of similar optimisation problems into several classes and subclasses with similar characteristics of fitness landscape measures, which define similar difficulties for the optimisation algorithm. Searching for better problem subclass specific algorithms and configurations will provide useful knowledge on the problem solution scenarios [37].

It would be convenient to represent a fitness landscape graphically in a visual image, but naturally it is possible for problems only with one fitness function and two continuous parameters (see Fig. 1). The fitness landscapes can always be described by a directed graph whose vertices are solutions labelled with fitness values, and edges are neighbour relations between relevant solutions [46].

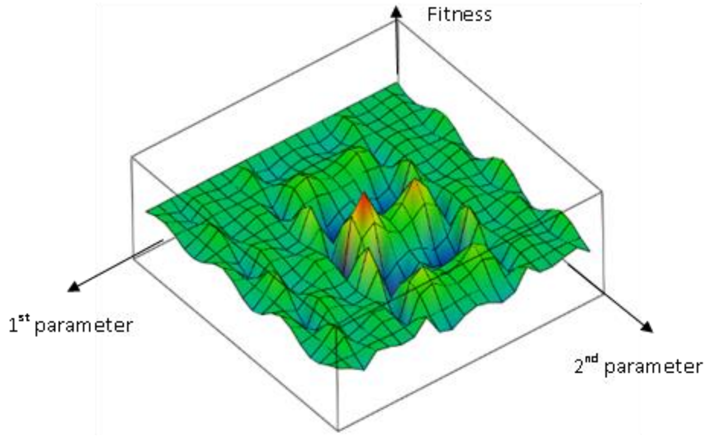


Fig. 1. Three-dimensional interpretation of fitness landscape

It should be noted that in the previous research on the landscapes, the behaviour of the algorithm in the problem's search space often relies on the intuitive geographical understanding of a fitness landscape. However, this conception often is not based on a formalisation and can be misleading [37].

To overcome the problems in the interpretation of a fitness landscape, formal definitions are known in the literature. The commonality for all these definitions is that they define a fitness landscape as the combination of the fitness function and the topological structure of the search space.

In the following formal definitions a notion of a multiset [25] is applied. The multiset $\boldsymbol{M}(S)$ is defined, which is an infinite set of multisets that contain elements of a multiset S. Moreover, $\boldsymbol{M}_q(S)$ is a set of all multisets that are derived from S and have cardinality equal to q [20].

An *object space* of the problem is defined as $\boldsymbol{O}$. A search algorithm has to select an appropriate object from all available $\boldsymbol{O}$ objects, which can be any type of structure, e.g. tuple, a set of permutations, etc. A *representation space* $\boldsymbol{R}$ defines a set of *representations* that represent objects from the set $\boldsymbol{O}$. The mapping between $\boldsymbol{O}$ and $\boldsymbol{R}$ is called *representation*.

The representation between sets $\boldsymbol{O}$ and $\boldsymbol{R}$ is defined by the relation Γ. For elements $o \in \boldsymbol{O}$ and $r \in \boldsymbol{R}$ the notation $o\Gamma r$ defines that *o is represented by r*. A reverse relation Γ^{-1} between sets $\boldsymbol{R}$ and $\boldsymbol{O}$ is defined as $\Gamma^{-1} = \{(r, o) \mid (o, r) \in \Gamma\}$, where the notation $r\Gamma^{-1}o$ defines that *r represents o*. If $\Gamma(o) \neq \emptyset$, then the object o is *represented*. If $\Gamma^{-1}(r) = \emptyset$, then the representation r is *illegal*.

The definition of a problem for which the landscape metaphor can be applied has some function $g : \boldsymbol{O} \rightarrow \boldsymbol{G}$ to determine a quality of solutions. Here $\boldsymbol{G}$ defines such a set with a partial order $>_{\boldsymbol{G}}$, for which, if $g(o_1) >_{\boldsymbol{G}} g(o_2)$ for $o_1, o_2 \in \boldsymbol{O}$, then o_1 is a better problem's solution than o_2.

At the time moment t algorithm operates on a finite number of $\boldsymbol{R}$ elements that form a multiset $\boldsymbol{C_t} \in \boldsymbol{M}(\boldsymbol{R})$. The current multiset will be denoted by $\boldsymbol{C}$. To modify the multiset $\boldsymbol{C}$, *operators* are applied by an algorithm. The operator will be denoted by ϕ, which is defined by a function ϕ: $\boldsymbol{M}(\boldsymbol{R}) \times \boldsymbol{M}(\boldsymbol{R}) \rightarrow [0, 1]$. A value of $\phi(v, w) = p$ for $v, w \in \boldsymbol{M}(\boldsymbol{R})$ defines a probability p that v will be modified to w by application of the stochastic procedure defined by the ϕ operator [20].

The search algorithm can be shown in a general way as the sequence of operators that modify a multiset of the selected solutions [20]:

$$C_0 \xrightarrow{\phi_1} C_1 \xrightarrow{\phi_2} C_2 \xrightarrow{\phi_3} \ldots \tag{1}$$

To define a landscape, the notation of ϕ-neighbourhood has to be defined for the operator ϕ. The ϕ-neighbourhood for the solution $v \in \boldsymbol{M}(\boldsymbol{R})$ is a set $N_\phi(v)$ of $\boldsymbol{M}(\boldsymbol{R})$ elements, which are accessible from v by one iteration of the ϕ operator. As a result, the set of neighbours for v is defined as:

$$N_\phi(v) = \{w \in \boldsymbol{M}(\boldsymbol{R}) \mid \phi(v, w) > 0\}. \tag{2}$$

If for a landscape point $w \in N_\phi(v)$, then w is a ϕ-neighbour of w. The ϕ in neighbourhood notation is essential, because ϕ-neighbour solutions may be not neighbours for other operators.

The neighbourhood $N_\phi(P)$ for a set of $P \subseteq \boldsymbol{M}(\boldsymbol{R})$ is defined as a set of such $\boldsymbol{M}(\boldsymbol{R}) - P$ elements, which are ϕ-neighbours of the elements of P:

$$N_\phi(P) = \{w \in \boldsymbol{M}(\boldsymbol{R}) - P \mid \phi(v, w) > 0 \text{ and } v \in P\}. \tag{3}$$

Certain goal function $f : \boldsymbol{M}(\boldsymbol{R}) \rightarrow \boldsymbol{F}$ on some set $\boldsymbol{F}$ with a partial order $>_{\boldsymbol{F}}$ defined on $\boldsymbol{F}$ has to be selected, in order to define which solution is the best in the current multiset of solutions, and to determine a search trend for the algorithm. In most cases a function f is taken relevant to the goal function g of the solved problem, so that $f = g$, $\boldsymbol{F} = \boldsymbol{G}$ and $>_{\boldsymbol{F}} = >_{\boldsymbol{G}}$. But this is not mandatory; often in application of genetic algorithms the function f includes a goal function of the problem g and some additional functions which penalize illegal objects.

The strict definition of the fitness landscape is given in the [20]. The landscape is defined as the 5-tuple:

$$\boldsymbol{L} = (\boldsymbol{R}, \phi, f, \boldsymbol{F}, >_{\boldsymbol{F}}), \tag{4}$$

where the components of the fitness landscape are as follows: a representation space $\boldsymbol{R}$, an operator ϕ that forms this landscape, a fitness function $f : \boldsymbol{M}(\boldsymbol{R}) \rightarrow \boldsymbol{F}$ which maps the multiset of a representation space on some set, called the fitness space $\boldsymbol{F}$, and a partial order $>_{\boldsymbol{F}}$ on $\boldsymbol{F}$.

The landscape here is the metaphor, which is applied to get more information on the behaviour of a search algorithm. The 5-tuple of the landscape can be represented as a directed labelled graph $G_{\boldsymbol{L}} = (V, E)$, where vertices are $V \subseteq \boldsymbol{M}(\boldsymbol{R})$, and where edges are $E \subseteq V \times V$ and the edge is defined if $(v, w) \in E \Leftrightarrow \phi(v, w) > 0$. In this representation, a vertex $v \in V$ is labelled as $f(v)$, and edge (v, w) is labelled $\phi(v, w)$. While the formal statement of the landscape is the 5-tuple, the landscape may be interpreted as the graph, which is obtained from the tuple [20].

In an application of this definition for describing the fitness landscape, it is not possible to say that a GA makes a walk on the landscape. GA operators make small steps on crossover and mutation landscapes, and then with the whole population on a selection operator's landscape.

A simplified definition of the fitness landscape is given in the study [37]. Here, the fitness function is defined as f: $\boldsymbol{R} \rightarrow \mathbb{R}$. The distance d between neighbour solutions is defined as $\boldsymbol{R} \times \boldsymbol{R} \rightarrow \mathbb{R}$, which forms a metric instead of a neighbourhood function N: $\boldsymbol{R} \rightarrow 2^{\boldsymbol{R}}$. The fitness landscape $\boldsymbol{L}$ is defined as a set of two functions f and d that define the fitness value and the distances between solutions in $\boldsymbol{R}$ as:

$$\boldsymbol{L} = (\boldsymbol{R}, f, d). \tag{5}$$

In (5) the operator and fitness space with its order are omitted, while the fitness function maps on a set of real numbers $\mathbb{R}$. The definition of the ε-neighbourhood for solution x in this notation is given by:

$$\mathrm{N}_{\varepsilon}^{+}(x) = \{n \mid n \in \boldsymbol{R}, n \neq x, d(x, n) \leq \varepsilon\}. \tag{6}$$

By requiring a distance measure in addition to the fitness value, the fitness landscape is not only dependent upon the problem, but also strongly linked to the choice of representation and its connectedness using certain operators for moving between or recombining solution candidates [37].

Furthermore, the definition in [33] states that it is possible to represent the fitness landscape of any optimisation problem as a graph $G_L = (V, E)$, where V is a set of vertices, which are mapping a set of solutions S, and $E = \{(s, s') \in S \times S \mid d(s, s') = d_{\min}\}$ is a set of edges, where d is a distance between two solutions, and d_{min} is a minimal distance between these solutions. This distance is defined by a number of applications of operator necessary to transform a selected solution to another one [33].

Similar to structures of nature landscapes hill ridges, valleys and other structures can be identified in the fitness landscape. Hills or peaks on the landscape are defined as solutions that have its fitness better than their neighbourhood. The peak that has the highest fitness is called as a global optimum, but all other peaks are called local optimums.

The following formal definitions of the fitness landscape structures are given for a case, when a landscape is defined in the form (4). These definitions are based on the definitions of a *ϕ-neighbourhood* and a *ϕ-neighbour*, which are given above.

A vertex is a *ϕ-peak* (or *ϕ-maximum*) only if its fitness is better than any fitness of its ϕ-neighbours [20] and is defined in the following way:

$$v \in V \mid \forall w \in N_{\phi}(v), f(v) >_{\boldsymbol{F}} f(w). \tag{7}$$

For a simplified form (5) and the ε-neighbourhood definition given in (6) the local optimum is defined in [37] as:

$$\text{local optimum}(x) :\Leftrightarrow (\exists \varepsilon > 0)\ (\forall n \in N_{\varepsilon}^{+}(x))\ f(x) > f(n). \tag{8}$$

A vertex is a *global-maximum* when its fitness is not worse than any other solution fitness on the whole landscape. Such vertex v of the landscape graph is defined by:

$$v \in V \mid \forall w \in V, f(v) >_{\boldsymbol{F}} f(w). \quad (9)$$

In turn, a *ϕ-local-maximum* or *ϕ-local-optimum* is such *ϕ-peak* that is not the ϕ-global-maximum at same time.

The region of landscape, where all solutions are with a similar fitness is called a *ϕ-plateau* and is a set of vertices that is defined by:

$$M \subseteq V, |M| > 1 : \forall \mathrm{v}_0, v_n \in M, \exists v_1,...,v_{n-1} \text{ for}$$
$$f(v_i) = f(v_{i+1}) \text{ and } v_{i+1} \in N_\phi(v_i) \;\; \forall\, 0 \le i < n. \quad (10)$$

The notion of a basin of attraction is used in the interpretation of the results of the fitness landscape analysis. This is a set of vertices from which a given vertex can be achieved by an iterative application of operator. *ϕ-basin-of-attraction* for a vertex v_n is formally defined as follows:

$$B_\phi(v_n) = \{v_0 \in V \mid \exists v_1, ..., v_{n-1} \text{ with } v_{i+1} \in N_\phi(v_i) \;\;\; \forall\, 0 \le i < n\ \}. \quad (11)$$

An operator ϕ has a fixed cardinality if it is defined in form $\boldsymbol{M}_k(\boldsymbol{R}) \times \boldsymbol{M}_l(\boldsymbol{R}) \to [0, 1]$ for certain indexes k and l. Further in the text notation $\phi_{k\to l}$ will be used to refer to such an operator. Operators that are of $\phi_{k\to k}$ are called *walkable operators*. Most of the mutation operators in genetic algorithms are walkable. An operator ϕ is called *symmetric*, if $\phi(v, w) = \phi(w, v)$ for any vertices $v, w \in V$. The majority of known operators of evolutionary algorithms are symmetric [20].

A landscape that is formed by an operator with a fixed cardinality is called *natural* if the operator is symmetric and is of $\phi_{1\to 1}$ class. The name “natural” comes from the basic interpretation of the fitness landscape [20].

Literature on fitness landscape analysis defines a number of factors, which affect the hardness of the optimisation problem and hence will allow selecting an appropriate metaheuristic algorithm and its configuration.

The major highlighted characteristics are the modality, ruggedness, neutrality, epistasis [39] and neutrality. The *epistasis* defines interaction between the genes in the chromosome, when the behaviour of a genotype is relevant on the combination of the gene interactions [4, 26, 39]. *Modality* defines a number and density of optima in a search space [39]. *Ruggedness* characterizes the impact of all structures of the landscape on the hardness of the search and the landscape is rugged, if there is small correlation between neighbourhood solutions [33, 48]. *Neutrality* characterizes a number of structures with a very similar fitness in the landscape [40, 43]. This measures a number and size of plateaus areas.

The following factors which influence the structure of the problem’s fitness landscape, and should be analysed while selecting, constructing or tuning the optimisation algorithm: problem hardness, solution representation, search operators and a fitness function. The hardness of the problem is the only factor, which cannot be changed without changing the statement of the optimisation problem.

Selection of different representations affects the representation space and topological structure of the landscape. Different properties and characteristics of the landscape are obtained if different encoding methods are applied [26]. As it was shown above, a search operator significantly influences the topological structure of the landscape. So, different fitness landscapes may be defined for different operators. Furthermore, choice of a fitness function doesn’t change the topology of the landscape, while it affects the behaviour of a search algorithm. For example, fitness function with many stochastic parameters can make the landscape noisier and thus more rugged.

Review of fitness landscape analysis techniques

A number of different techniques have been developed for a fitness landscape analysis by evaluating its structural characteristics. Fitness landscape analysis techniques provide the possibility to analyse the structure of the fitness landscape by analysing only small part of the landscape. A number of fitness landscape analysis techniques and measures are defined in the literature [6, 8, 16, 18, 21, 43, 44, 46, 47, 48]. These techniques can be divided into two groups that provide a statistical and information analysis of the landscape. Statistical analysis techniques use statistical data and evaluate correlations between potential solutions on the fitness landscape. Information analysis techniques use measures from information theory to estimate structural features of a fitness landscape.

Fitness landscape analysis techniques do not require information about all problem solution candidates, but analyze only a part of a fitness landscape data and apply different strategies for data collection. Techniques described in this thesis are based on simple moves. A sequence of such moves generates a trajectory through the landscape, which is then analysed. Most methods for a fitness landscape analysis are based on a random walk. Other landscape walks such as up-down (adaptive) and neutral walks extend the amount of information collected on the landscape and allow for getting additional characteristics. The above mentioned three types of landscape walks, which are most known in literature, are applied in this thesis.

In the *Random Walk*, a solution candidate is randomly modified repeatedly, and a random trajectory on the landscape is created. In the *Adaptive* landscape walk, a certain number of mutations are performed to generate a set of neighbours, and then the best one is selected from this set [22]. The *Up-Down Walk* is similar to the adaptive walk, but when the up-down walk reaches a local optimum, the direction of the walk is reversed [48]. *Neutral Walks* explore “flat” areas. Here, neighbours with an increasing distance from a certain starting point are chosen while trying to remain at the same fitness level as the starting point [40].

The ***statistical analysis***, which is proposed in [50], is a widely used approach for the fitness landscape analysis. This approach basically calculates the autocorrelation function in the random walk. Within the statistical analysis, the correlation between fitness of neighbourhood solution candidates is used to measure the ruggedness of the landscape. In case of a high correlation between two sets of fitness values of landscape points the fitness landscape is considered smooth or less rugged.

In the first step of the statistical analysis a long walk is performed on the landscape, and the collected fitness values form time series $\{f_t\}^N_{t=1}$. Then, an autocorrelation function is calculated for the collected time series. The autocorrelation function $\rho(\Gamma)$ indicates the correlation between two sets of points that are separated by a distance Γ [46]:

$$\rho(\Gamma) \approx \frac{E(f_t f_{t+s}) - E(f_t)E(f_{t+s})}{V(f_t)}, \tag{12}$$

where $E(f_t)$ the expectation and $V(f_t)$ is the variance of a sequence of fitness values $\{f_t\}_{t=1}^N$, where N is the length of the sequence. For smooth landscapes the autocorrelation of a random walk is close to 1 and tends to zero for rugged landscapes [39].

Another statistical measure is correlation length, which defines a distance beyond which two sets of fitness points becomes uncorrelated. The correlation length in [50] is evaluated by:

$$\tau = \frac{1}{\ln(\rho(1))}, \tag{13}$$

where $\rho(1)$ is the autocorrelation of the neighbouring points. A longer correlation length indicates a smooth landscape, while a shorter length indicates a rugged landscape. More robust evaluation of the correlation length for the fitness landscape analysis is proposed in [18]. The model of autocorrelation is extended by the definition, where correlation is significant while it exceeds two standard-error-bounds $\left(-2/\sqrt{N}\,;+2/\sqrt{N}\right)$, where N is the length of the time series. The correlation length τ is one less than the first time lag, where the autocorrelation $\rho(\tau+1)$ falls in the error bound [18].

The ***information analysis*** interprets a fitness landscape as an ensemble of various objects, which are characterized by their form, size and distribution. In [48] these objects consist of a point in the fitness landscape and the nearest neighbours of the point. The information analysis is based on the information theory, and four information measures are proposed in [48].

All information measures are calculated with notice to a calculation accuracy which is defined by parameter ε. This parameter defines a threshold of slopes in the fitness path. All slopes that have fitness difference between the neighbour solutions less than ε are assumed to be flat [48]. The information content $H(\varepsilon)$ is a measure of entropy in the system. In case of high information content, the landscape has a large variety of structures and is more rugged. Partial information content $M(\varepsilon)$ characterizes the modality of the obtained fitness string [46]. The information stability ε^* characterizes a magnitude of optimums in the obtained landscape fitness path. The density-basin information $h(\varepsilon)$ analyses the variety of flat and smooth sections in the landscape.

To obtain the information measures of the landscape $\boldsymbol{L}$, the landscape walk is performed, and fitness values of passed solutions are collected in the time series $\{f_t\}^N_{t=0}$, where N is a length of the random walk. The sequence of fitness values is transformed into a string of ensembles $S(\varepsilon)$, which elements $s_i \in \{\bar{1},0,1\}$ are calculated by:

$$s_i(\varepsilon) = \begin{cases} \bar{1}, & if \quad f_i - f_{i-1} < -\varepsilon, \\ 0, & if \quad |f_i - f_{i-1}| < \varepsilon, \\ 1, & if \quad f_i - f_{i-1} > \varepsilon. \end{cases} \tag{14}$$

The parameter ε is a real number between 0 and highest difference between fitness values in the landscape path and defines how accurate the string $S(\varepsilon)$ is calculated.

The information content $H(\varepsilon)$ is defined by:

$$H(\varepsilon) = -\sum_{p \neq q} P_{[pq]} \log_6 P_{[pq]}, \tag{15}$$

The partial information content $M(\varepsilon)$ is determined by:

$$M(\varepsilon) = \frac{\Phi_s(1,0,0)}{n}, \tag{16}$$

where n is the length of the string $S(\varepsilon)$, and the function Φ is calculated recursively by:

$$\Phi_S(i,j,k) = \begin{cases} k & ,if \quad i > n; \\ \Phi_S(i+1,i,k+1), & if \quad j = 0 \quad and \quad s_i \neq 0; \\ \Phi_S(i+1,i,k+1), & if \quad j > 0, s_i \neq 0 \quad and \quad s_i \neq s_j; \\ \Phi_S(i+1,j,k) & ,otherwise. \end{cases} \tag{17}$$

Finally, the density-basin information $h(\varepsilon)$ is determined by:

$$h(\varepsilon) = -\sum_{p \in \{1,0,\bar{1}\}} P_{[pp]} \log_3 P_{[pp]}, \tag{18}$$

where the probabilities $P_{[pp]}$ represent frequencies of sub-blocks pp from the string $S(\varepsilon)$ [48].

Information stability ε^* is the lowest ε value, with which it is obtained that the fitness path has no structures at all.

The fitness landscape of a problem that is easy for evolutionary algorithms has values of information measures close to 0. For hard optimisation problems these values are close to 1 [48].

The statistical and information analysis techniques can be used only for statistically isotropic fitness landscapes. For these landscapes a fitness value sequence obtained with a random walk forms a stationary random process for the assumed joint distribution of the fitness values [48].

Problem setup of the simulation-based fitness landscape analysis and optimisation

To extend the concept of the fitness landscape analysis for its application in simulation-based optimisation, the concepts of the simulation fitness landscape and simulation-based fitness landscape analysis are introduced. The formal definition of the simulation fitness landscape $\boldsymbol{L}'$, which is derived from a modification of definition (4) by substituting the fitness function f with a simulation model, follows:

$$\boldsymbol{L}' = (\boldsymbol{R}, \phi, S, \boldsymbol{F}', <_{\boldsymbol{F}'}), \qquad (19)$$

where $\boldsymbol{R}$ is a representation space, ϕ is a search operator, S is a simulation model, $\boldsymbol{F}'$ is a set of a simulation model outputs that has an order defined by $<_{\boldsymbol{F}'}$. With an assumption that a simulation model provides real value output, the definition (19) of the landscape can be translated into the following definition:

$$\boldsymbol{L}' = (\boldsymbol{R}, \phi, S), \qquad (20)$$

where $\boldsymbol{R}$ is a representation space, ϕ is an operator that forms the landscape and S is a simulation model with one output variable:

$$S = f : \vec{x} \times \xi \mapsto \hat{y}, \qquad (21)$$

where $\vec{x} \in \boldsymbol{R}$ is a vector of simulation model input variables, which represent solution candidates in the representation space, f is an objective function, ξ is a random component of the simulation model and $\hat{y} \in \mathbb{R}$ is the mathematical expectation of the simulation model output.

To apply the fitness landscape analysis in the simulation optimisation, the following three-level formalised scheme is introduced (see Fig. 2), which contains the benchmarking level or level-0, and two levels of landscape analysis and optimisation, correspondingly.

At the *benchmarking level*, information on different landscape structures and measures and on the performance of the optimisation algorithms on benchmark landscapes is collected. At the *landscape analysis level*, the landscape analysis procedure is defined. First, walks on the landscape are performed with different walking strategies, and the trajectory on the landscape defined by a simulation model and selected operator is generated. The time series of fitness values obtained in the walks are collected and landscape analysis measures are calculated by using statistical and information analysis techniques. The obtained collection of data with the information on walking strategies and landscape measures is used to select and adjust an appropriate optimisation algorithm. Moreover, collected data is also added to a dataset of benchmarking landscapes that are useful for future cases. At the *optimisation level*, the selected algorithm is used to optimise the investigated system by using the simulation-based metaheuristic optimisation approach. Information on the performance of the optimisation algorithm is collected and added to a dataset of benchmarking landscapes.

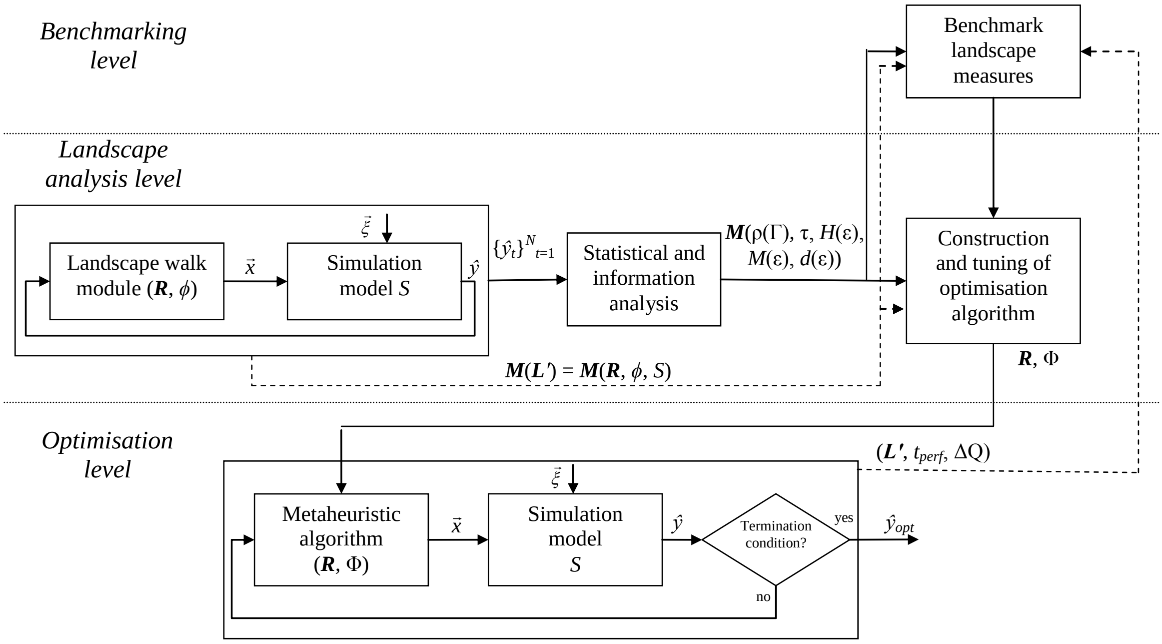


Fig. 2. Simulation-based optimisation with fitness landscape analysis

Landscape walk module *LW* can be interpreted as follows:

$$\vec{x}_{t+1} = LW(\vec{x}_t, \phi, \hat{y}), \qquad (22)$$

where t is a number of landscape walk iterations completed from N expected iterations in the walk, $\vec{x}_{t+1}$ is a vector of simulation model input variables for a current iteration, $\vec{x}_t$ are input variables at the previous iteration, ϕ is a search operator and $\hat{y}$ is an output of a simulation model. For the landscape walk, an operator ϕ is applied with notice to the information on the representation space $\boldsymbol{R}$. The output $\hat{y}$ of a simulation model is used in up-down and neutral walks in order to determine walking direction. Only walkable operators can be applied in this definition.

Output of the landscape walk module is a vector of simulation model input variables $\vec{x} = (x_1, x_2,..., x_k)$, $\vec{x} \in \boldsymbol{R}$, where $\boldsymbol{R}$ is a representation space and k is a number of simulation model input variables. For each input variable it is true that $x_i \in \mathbb{R}, \forall\ 0 \leq i < k$.

Simulation model S is used to evaluate the performance of a system to be optimised. It produces the simulation output from several model runs or replications and is defined by:

$$S = f:\ \vec{x} \times \vec{\xi} \mapsto \hat{y}, \qquad (23)$$

where f is an objective function, which optimal value is searched in the optimisation process;

$\vec{x} = (x_1, x_2,..., x_k)$ is a vector of k input variables;

$\vec{\xi} = (\xi_1, \xi_2,..., \xi_d)$ is a disturbance vector of d environmental variables;

$\hat{y}$ is a mathematical expectation of simulation output.

The output of the simulation model S is estimated by $\hat{y} = \mathrm{E}[y]$, where $y \in \mathbb{R}$ defines simulation output in each replication, and $\mathrm{E}[\cdot]$ denotes the mathematical expectation. Here, the simulation model is interpreted as a black box which defines input-output relationships of the model, not considering the states of the simulation model.

As a result of process integration of landscape walk LW module and simulation model S, a number of time series $\{\hat{y}_t\}^N_{t=1}$ in landscape walks is generated, where N is a number of evaluations in the trajectory. Lets note that the time series $\{\hat{y}_t\}^N_{t=1}$ is used as inputs for landscape analysis techniques, where it is denoted as $\{f_t\}^N_{\mathrm{t}=1}$ below.

The module of **statistical and information analysis (**S&IA) performs analysis of sequences of fitness values $\{f_\mathrm{t}\}^N_{t=1}$ obtained in the landscape walk, and calculations of the landscape statistical and information analysis measures are performed in this module. A set of values $\rho(\Gamma)$, τ, $H(\varepsilon)$, $M(\varepsilon)$, $h(\varepsilon)$ are obtained for different autocorrelation distance Γ and different sensitivity value ε. The module *S&IA* can be interpreted as:

$$S\&IA : \mathbb{R}^N \times \mathbb{N} \times \mathbb{R}_+ \to \mathbb{R}^5$$

$$(\{f_\mathrm{t}\}^N_{t=1}) \times \Gamma \times \varepsilon \mapsto (\rho, \tau, H, M, h). \qquad (24)$$

The module of **construction and tuning of an optimisation algorithm** allows selecting the appropriate optimisation algorithm and adjust its parameters for optimisation of a complex system, which is simulated by S. Selection of the algorithm, its components and parameters is based on the data from simulation-based fitness landscape analysis and the data on benchmark landscapes. The main input of this module is a set of values that describe previously analysed fitness landscape. The module output defines the selected metaheuristic optimisation algorithm and its configuration. The last one includes the representation of the solutions which defines the representation search space $\boldsymbol{R}$, and a set Φ of search operators which form the metaheuristic optimisation algorithm. The selection of the algorithm and its configuration is based on the rules and recommendations which are applicable for known values of the landscape measures. The alternative way of the algorithm selection and configuration is based on data of benchmark landscapes, where better optimisation algorithms are associated with corresponding landscape measures.

At the *optimisation level,* the **simulation model** is defined by the (23). Here, the **metaheuristic algorithm** (*MA*) here uses the representation method (defined by $\boldsymbol{R}$), a set Φ of suggested operators which are determined in the construction and tuning module. The *MA* module can be formalized here as follows:

$$MA:(R,\Phi,\hat{y},M(\langle\vec{x},\hat{y}\rangle)_t)\mapsto\vec{x}_{t+1}, \qquad (25)$$

where t is a number of performed evaluations of a simulation model;

$\vec{x}_{t+1}$ is a vector of suggested simulation model input variables;

$M(\langle\vec{x},\hat{y}\rangle)_t$ denotes a memory of previous solution candidates obtained after the t-th evaluation;

$\hat{y}$ is a mathematical expectation of a simulation output.

The **termination condition** determines whether the suitable solution is found and an optimisation cycle can be stopped. When the optimisation cycle is terminated, the best found solution $\hat{y}_{opt} = \langle\vec{x},\hat{y}\rangle$ is selected. The performance measures of the optimisation model can be added to the dataset of benchmark landscape measures. These measures are presented by a three-valued tuple including the problem landscape (20); time performance measure t_{perf} of the optimisation algorithm which determines time required to find the best known solution; and performance measure ΔQ which determines quality improvements for the best found solutions.

Here, the simulation-based fitness landscape analysis is a way to examine the fitness landscape generated on the simulation model basis, providing an overview of how the optimisation module has to be configured and tuned.

2. Benchmark Fitness Landscape Experimental Analysis

This section presents implementation of fitness landscape analysis techniques to estimate measures of fitness landscapes for several benchmark functions with known structures defined by analytical expressions. Also, the section provides an experimental analysis on how random noise in the fitness function can affect measures of fitness landscapes. Detailed analysis of benchmark fitness landscape structures is given in order to find the relevance between landscape measures, landscape factual structures and behaviour of optimisation algorithms.

Benchmark fitness landscapes

The following four fitness functions are widely used for benchmarking of genetic algorithms and were selected for estimating and analysing statistical and information measures of benchmark fitness landscapes. They are Sphere function [9], Rastrigin function [53], Rosenbrock function [9] and Ackley function [1]. These functions can be defined in the same search domain with an equal number of variables and can easily be graphically interpreted for two variables.

The Sphere function is a continuous, convex, quadratic and unimodal, and for the vector $X = \langle x_1,...,x_n\rangle$ of n variables it is defined by:

$$f_{Sphere}(\langle x_1,...,x_n\rangle)=\sum_{i=1}^{n}x_i^2. \qquad (26)$$

The Rastrigin function has more rugged landscape and is defined by:

$$f_{Rastrigin}(\langle x_1,\ldots,x_n\rangle)=10n+\sum_{i=1}^{n}\left(x_i^2-10\cos(2\pi x_i)\right). \qquad (27)$$

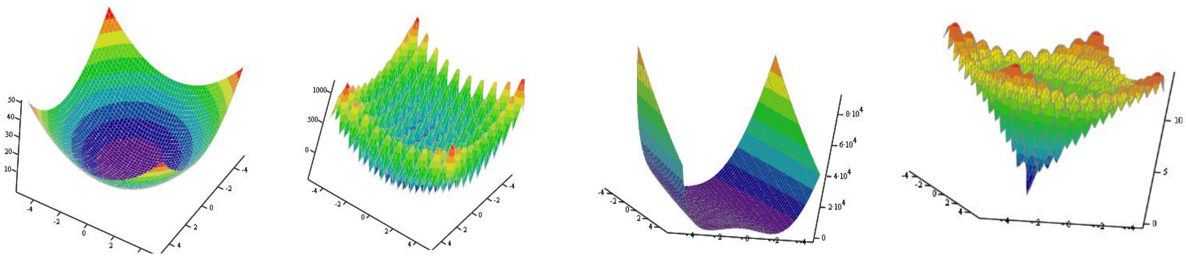
The Rosenbrock function is a continuous, non-convex, quartic and unimodal function of n variables and is calculated by:

$$f_{Rozenbrock}(\langle x_1,\ldots,x_n\rangle)=\sum_{i=1}^{n-1}\left((1-x_i^2)+100(x_{i+1}-x_i^2)^2\right). \qquad (28)$$

Finally, the Ackley function is highly multimodal and rugged. Like the Rastrigin function, it has the local optima, but the slope to the optima is exponential. The Ackley function is defined by:

$$f(\langle x_1,\ldots,x_n\rangle)=20+e-20\exp\left(-0{,}2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right). \qquad (29)$$

Sphere, Rastrigin and Rosenbrock functions have the global optimum in the point $x_i = 0$, $i=1...n$, but for the Rosenbrock function it is in the point $x_i = 1$, $i = 1...n$, located inside a parabolic shaped flat valley. Visualisations of all four functions for 2-dimensional problems are shown in Fig. 3.



Sphere function   Rastrigin Function   Rosenbrock function   Ackley function

Fig. 3. Visualisation of landscapes of benchmark functions

For all four functions, a number of variables is taken equal to $n = 2$, and a search domain is defined by $-5 \le x_i \le 5$, $i = 1, 2$. Two types of solution representations are used: real-value encoding and binary representation. In the first, variables x_1 and x_2 are coded as real numbers with a resolution factor of 0.01, e.g. –3.49 and 0.84. Binary coded chromosomes have length of 20 bits, where first 10 bits code x_1, but others code x_2.

Different mutation operators are applied for different encodings. For real-value encoding, a mutation operator changes each variable in a chromosome by +0.01 or –0.01 with the probability equal to 1/3. If all variables x_i stays unchanged, the mutation operator is re-applied until at least one variable in this chromosome changes its value. For binary representation, a bitflip operator is used, which changes the value of a randomly selected bit to the opposite value.

Therefore eight different fitness landscapes received for four different benchmark functions and two types of a search space for each function are analysed.

Experimental analysis of benchmark landscapes

For the Sphere, Rastrigin and Rosenbrock fitness landscapes, an express analysis was performed to show and describe main steps of the landscape analysis. For a detailed analysis of benchmark fitness landscapes, a software prototype in Java was developed and applied. To estimate structural measures of these landscapes, multiple experiments were performed with multiple randomly generated paths and a starting point uniformly distributed in the representation space.

Autocorrelation functions calculated for different benchmark functions and lags demonstrate similar results. For longer trajectories on the landscape the values of the autocorrelation function are closer to 1. While correlation measures show dependence on the length of the path generated by a random walk, the behaviour of information content measures does not demonstrate this effect.

In the second series of experiments, the autocorrelation for different benchmark landscapes and lags was defined for two types of solution representation. Correlograms obtained for real-value and binary coded benchmark landscapes show the higher autocorrelation for real-value coded fitness landscapes that make search processes easier in practice. However, the difference between correlogramms for different landscapes with the same representation type is minor.

In the third series of experiments, different information measures for all benchmark landscapes and different ε values were estimated. The results show the higher information content for smaller values of parameter ε; with an explicit peak around $\varepsilon = 0.03$ for the real-value coded Sphere function. At $\varepsilon = 0$ information measures become identical and essentially do not provide a new information about structures of specific fitness landscapes. In this case, the information content tends to 0.388, the partial information content to 0.5 and the density-basin information to 0.63. At the same time, smaller values of the information content for the Rosenbrock function compared to the Sphere indicate the higher degree of flatness with respect to rugged areas of the landscape.

Due to different ranges of fitness value, it is difficult to compare information measures when $\varepsilon > 0$. To overcome this, value ε_1 that defines a similar and comparable part of the interval of all fitness values of the given landscape was specified. For Sphere, Rastrigin and Ackley functions it was taken as $\varepsilon_1 = 0.04$, while for Rosenbrock function $\varepsilon_1 = 50$, as this function has higher changes between the fitness of neighbour solutions.

Multiple experiments were performed for three small areas on the Sphere benchmark landscape near the global minimum, close to the local maximum and between them. A starting point in a random walk in each local search was defined by <0.05;–0.01>, <4.49; 4.82> and <–3.71; 1.23>, correspondingly. Results show that the autocorrelation is not significantly sensitive to a starting point. The magnitude of fitness change during the walk between neighbouring points has a big impact on the information content measures. Therefore, the information content shows a high sensitivity to a parameter ε so that ε values need to be carefully defined for different local areas of the landscape.

Noise and measures of landscapes

Additional experiments were performed to define the effects of random noise in a fitness function on statistical and information measures. In this case, the fitness function f^* is described as follows:

$$f^* = f + \xi, \tag{30}$$

where f is the benchmark fitness function and ξ is a term that represents noise effects and is treated as a normally distributed statistical error with a mean of zero and variance σ^2.

The results of experiments show that both statistical and information measures are quite sensitive to a noise factor. With increase of variance, the autocorrelation gets lower for shorter lags and higher for longer lags. Random noise increases the values of the information content correspondingly to an increase of the entropy of the landscape structures.

GA optimisation experiments with benchmark fitness functions

To find the correlation between the results of fitness landscape analysis and hardness of a real problem for an evolutionary algorithm, a series of optimisation experiments were performed with benchmark fitness functions. GA with one point crossover and above described mutation operator was used to estimate a cumulative probability of success (CPS) [23] for different benchmark landscapes. The results of optimisation experiments show that for binary coded Sphere, Rastrigin and Ackley functions, the global optimum is found in about 50-60% runs of more than 20 generations. The CPS doesn’t become higher for a larger number of generations. In the case of the Rosenbrock function, the global optimum is found in 10-20% runs.

In most cases, except for the Rosenbrock function, GA found solutions on real-value coded benchmark landscapes are better than on the binary ones that was predicted within the statistical analysis. As the autocorrelation between neighbourhood fitness points is high, it is easier for the genetic algorithm to move to a point with better fitness. Nevertheless, optimisation experiments show that for applied GA the Rosenbrock function is harder than the three other benchmark functions. This could be explained that existing fitness landscape analysis techniques are not able to identify structures that make search with GA difficult. To find out the reasons for the GA performance, dynamics of populations in different generations were analysed.

Distribution of found solutions in populations in different generations in the search space is compared for real-valued representation. For Rosenbrock function an experimental analysis shows that the population tends to a centre of the valley (Fig. 3), but not to the global optimum. This tendency was not observed for the GA with binary encoding, where the population converges toward the global optimum. For the other three functions, GA converges toward the global optimum much better for both types of encoding.

Cluster analysis for comprehensive fitness landscape analysis

In [38], the application of the fitness landscape analysis is performed in order to determine in what way problem instances of the same class (e.g. vehicle routing problems) are similar, or different. To define this, all values of the landscape measures which are obtained by different operators, are joined in one vector of problem instances characteristics. With application of this vector it is possible to make a cluster analysis of the problem instances and to define groups of similar instances.

Detailed analysis of the benchmarking landscapes

For the better interpretation of behaviour of the search algorithm on the benchmarking landscapes, their graphs of fitness landscapes were created and analysed. One solution candidate for Sphere function optimisation was randomly selected. Afterwards, the neighbourhood of the selected solution was constructed for different operators. The analysis of the real-valued mutation operator shows that solution candidates are highly connected with each other in this landscape, and the operator does not provide large changes for a fitness value. To represent a graph of a crossover operator, each vertex of the graph represents a pair of solution candidates [20]. The landscape graph of a crossover operator for real-valued representation is not connected. It is highly ineffective, as for each pair of solutions there is only one another pair which can be obtained with this crossover. Moreover, without the mutation operator this crossover would be non-productive as it does not generate a high number of new solutions from existing population.

The binary mutation operator, which landscape is a hypercube allows moving from one solution to a large number of other solutions. It is possible to move from the selected solution to another one with much better fitness only in two iterations of this mutation operator, but the probability of this move is very small. The binary crossover landscape is non-connected, but its each component consists of more vertices, and each solution has more neighbours. As the landscape of the binary crossover is walkable, three operator-local-optima are allocated in the corresponding component of the landscape graph. Thus, the fitness landscape of the binary crossover operator may be highly multimodal.

3. Simulation-based Fitness Landscape Analysis

Prototyping of fitness landscape analysis tool

Fitness evaluation of the potential solution is made through simulation. The procedure for the simulation-based fitness landscape analysis contains the following stages (Fig. 4):

1. Fitness landscape's path generation;
2. Fitness evaluation of solutions in the path;
3. Analysis of the path's fitness sequence.

To generate trajectories on the landscape and analyse the obtained fitness sequences, Java applications are developed. For fitness evaluation, a simulation model is developed in AnyLogic 6 simulation software. A standalone application performs a random walk on the problem fitness landscape. As a result, a sequence of parameters $\{\vec{x}_t\}_N^{t=1}$ of landscape path candidate solutions is obtained.

The simulation model is evaluated in AnyLogic *parameter variation* experiment with different input parameters, which are defined in the obtained trajectory. As a result, the model generates an array $\{\hat{y}_t\}_N^{t=1}$ of fitness values.

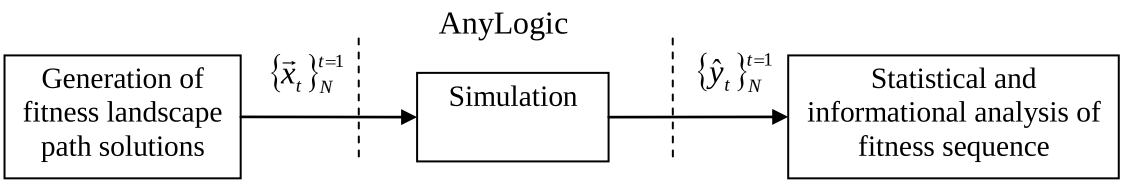


Fig. 4. Main stages of fitness landscape analysis

Finally, calculation of statistical and information fitness landscape measures is performed on a sequence of obtained fitness values. In statistical analysis, correlation length and autocorrelation function for different lags are calculated. In the information analysis three steps are performed:

1. Determination of information stability ε^*.
2. Iterative calculation of information content $H(\varepsilon)$, partial information content $M(\varepsilon)$ and density-basin information $h(\varepsilon)$ for different values of ε.
3. Output of values of information measures.

To define ε^*, an interval of possible ε values is divided by 2 in each iteration. A half-interval that contains the possible value ε^* is selected for further analysis in the next iteration. Measures $H(\varepsilon)$, $M(\varepsilon)$ and $h(\varepsilon)$ are calculated iteratively in the interval $[0, \varepsilon^*)$ with a step $0.05 \cdot \varepsilon^*$.

Experimental data from several random walks is collected during analysis. As all random walks are started at different random positions, fitness landscape measures are obtained for a large part of the landscape.

Case study

A case study based on the vehicle scheduling problem with time windows has been developed for validation of the fitness landscape analysis tool. Vehicles with various parameters deliver different types of goods from one distribution centre to many shops (customers). Distribution routes for vehicles are known. For each trip a sequence of shops in a route, average time intervals for vehicle moving between route points, as well as loading and unloading average times, and types of goods that can be carried in this trip are defined. Goods have to be delivered in shops only in the predefined time windows. An average demand of goods of each type is defined for each shop. Vehicle capacities are limited and known. The problem is aimed at assigning vehicles to trips in order to minimise the total idle time of all vehicles, which is defined as a sum of time periods when a vehicle is waiting for the next trip while in the depot.

Vehicle scheduling problem (VSP) is frequently reviewed in the studies [11, 35]. However, it is often modified with additional constraints. A number of methods to solve the VSP are proposed such as integer programming, heuristics, etc. However, all available solutions are not usable in many real life problems [11]. In practice, VSP can also be complicated by stochastic processes, and this is a reason to apply simulation optimisation to solve such problems.

The following two sets of decision variables are introduced for the vehicle scheduling problem in the thesis: vi and ti, where i is a trip number, vi is number of vehicle assigned to trip i and ti is start time of trip i.

The problem constraints are divided into two groups such as vehicle capacity constraints and delivery time constraints.

The objective function f is aimed to minimise the total idle time for all vehicles. Experimental analysis shows that the problem has many solutions which are not feasible. To improve the efficiency of optimisation, the objective function was modified, taking into account an amount of constraints that were not satisfied by a solution candidate:

$$f^{*}=\sum_{i=1}^{N} T_{idle}^{i} + k_1 T_c + k_2 T_m + k_3 T_0 + k_4 N_{ol} + k_5 N_{ot} \rightarrow \min, \qquad (31)$$

where f^* is the modified objective function; T_{idle}^{i} is the total idle time for vehicle i; N is a number of vehicles; T_c defines the total duration of overlapping trips for one vehicle; T_m defines the total time of window mismatches; T_o and N_{ol} determine the total time and a number of vehicles that have overdone 24 working hours; and N_{ot} is a number of vehicles that are overloaded. In (31), all indexes for unsatisfied constraints are multiplied with coefficients $k_i > 1,\ \ i = 1,\ldots,5$ that artificially increase a value of objective function and make the fitness of infeasible solutions worse.

Development of simulation model and sensitivity analysis

To determine the fitness of potential vehicle schedule solutions, discrete event simulation model in AnyLogic is developed, in which distribution centre specific processes are simulated. During a simulation process, constraint violations, such as time window mismatch in delivery and shortage of vehicles, are determined. AnyLogic simulation software is powerful tool for simulation model development and combines three simulation methodologies: system dynamics, discrete event and agent-based modelling.

The main task of the simulation model is to evaluate the efficiency of a potential vehicle schedule by estimating the average total idle time of all vehicles, taking into account stochastic conditions. As control variables, the parameters of the vehicle schedule are introduced. The simulation model is defined by two active objects (submodels): *main* object and *vehicle* object. The *main* object includes decision variables, functions for model's input data initialisation, variable collections of data of trips and shops, variables for total idle and total usage times of vehicles. The active object *vehicle* simulates vehicle processes using a state chart that defines vehicle's possible states and transitions between them. The *vehicle* object contains the job list of the corresponding vehicle in form of variable collection, variable for accumulation of idle time and variables which count a number of constraint unsatisfactory cases. Assignment of the vehicles to the trips in the simulation model is transformed to the assignment of jobs to a vehicle, where job is a combination of the trip and start time. Initialisation function of the simulation model transforms values of control variables to a scheduled list of jobs for each vehicle.

In a visualisation of vehicle processes, statistics of vehicle's utilisation is shown in a timeline chart, where different states of vehicles are shown with different colours. In a model animation the timeline charts of all vehicles are joined together in a Gantt chart of the vehicle schedule. The Gantt chart is combined with a chart of the actual usage of distribution centre gates during the daytime.

To validate the simulation model the existing vehicle schedule of a case study was simulated. The obtained schedule and its parameters correspond to the Gantt chart, received from a real-life vehicle schedule.

An analysis of how model's stochastic factors affect its output variables was performed. In this analysis, the vehicle moving time between two route points was defined as a random variable with normal distribution which mean value is equal to the mean moving time and the variance is proportional to the mean moving time, with different multiplication coefficient in different experiment series.

In experiments it was determined that stochastic nature of the vehicle moving time has an impact on the vehicle idle time which grows with the growth of variance. At same time, a sum of all moving intervals for a vehicle is not affected by variance of moving times.

Optimisation scenarios

To solve the vehicle scheduling problem with time windows, three optimisation scenarios are defined:

1. Optimisation in commercial OptQuest optimisation tool.
2. Simulation-based fitness landscape analysis and optimisation of the problem in the developed prototype.
3. Fitness landscape analysis and optimisation of the problem in the HeuristicLab framework.

As the simulation model of the vehicle schedule is developed in the AnyLogic software, optimisation tool OptQuest embedded in AnyLogic was applied in the optimisation. This tool uses scatter search, tabu search and additional heuristics, but embedded methods are poorly described in the literature. In experiments with OptQuest, it was not possible to obtain good solutions of the vehicle scheduling problem. Obtained solutions don't satisfy all defined constraints, even for the task with a reduced number of trips. However, application of this tool provided improvement of the current solution, which suggests that future improvement of the schedule is possible.

Problem research with developed simulation-based tools

In Scenario 2, the vehicle scheduling problem was sequentially analysed by the developed simulation-based fitness landscape analysis tool and optimised in simulation-based optimisation by tuned genetic algorithm.

Here, solution of the VSP is encoded as an integer vector chromosome, which length is twice the number of trips. Genes with even numbers represent start times of corresponding trips in minutes from midnight, and odd genes define the assigned vehicle for this trip.

To perform the random walk on the fitness landscape, a mutation operator is introduced that changes one randomly selected trip in the solution candidate. The probability of each trip to be changed is equal for all trips. For the selected trip a new randomly chosen vehicle is assigned, and start time is shifted by certain constant value.

The fitness landscapes of the VSP with stochastic and deterministic input data were analysed experimentally. In each series, 5 experiments with landscape's 100 solutions' long path were made. Information measures and statistical measures of the VSPTW fitness landscape received from the simulation experiments are given in Table 1. The sensitivity value ε for the calculation of landscape information measures was taken as 1/10 of the difference between the highest and lowest fitness values in the analysed path.

Table 1

Information and statistical measures

Model input data	$H(0.1)$	$M(0.1)$	$h(0.1)$	ε*	ρ(1)	ρ(10)	τ
Stochastic	0.66	0.20	0.49	0.40	0.84	0.21	7.24
Deterministic	0.62	0.17	0.37	0.35	0.89	0.32	8.75

Information measures demonstrate that the landscape of problem with stochastic data has higher entropy and should have higher modality. The information content is relatively high, and fitness landscape of the optimisation problem is relatively rugged. The partial information content is low, and as a result, the modality of fitness landscape is low. The density-basin information indicates that peaks have high density, and their density is higher for a stochastic problem.

According to the landscape measures, problem with stochastic data should be more complex for the optimisation algorithm as values of its autocorrelation function between neighbour solutions ρ(1) are lower, but correlation length τ is less than 8 solutions.

The results of the fitness landscape analysis in Scenario 2 lead to a conclusion that the application problem is not hard for evolutionary algorithms. Results of additional fitness landscape analysis experiments of benchmark fitness functions show that landscape statistical measures of the vehicle scheduling problem with time windows (VSPTW) are close to the corresponding measures of the Ackley function [1] in real-valued search space. Comparative analysis shows that landscape of VSPTW is less rugged than landscapes of benchmark fitness functions [31] whose solutions are coded in binary chromosomes. Thus the analysed problem could be solved with the GA no worse, than mentioned benchmark problems.

In simulation optimisation, the genetic algorithm is applied to search for the best combination of the schedule's parameters. The optimisation algorithm is based on the classical GA. The optimisation tool is implemented as a Java class, which interacts with the simulation model via 'Parameter variation' experiment in AnyLogic.

The considered simulation model has large dimensions, and the number of trips, vehicles and shops are equal to 37, 17 and 36, respectively. The total number of potential solutions is equal to $N = (17 \cdot 134)^{37} \approx 1{,}69 \cdot 10^{124}$. 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. Fitness function (31) is used in evaluation of potential solutions. Loading, moving and unloading times are defined by their mean values.

As simulation optimisation experiments are time consuming, caching of fitness values was applied in optimisation of the problem with deterministic data. Fitness values with the corresponding vector of decision variables after each simulation run are added to a special array. Before a new simulation run individuals are compared with members of this array, and if the next solution was already evaluated its fitness value can be returned without simulation.

Chromosomes are implemented as strings of integer numbers that encode parameters of the vehicle schedule. Each population contains 200 chromosomes. All genetic operators are customized for operating with the proposed structure of the chromosome. In particular, one point crossover operator for data encoded in real numbers is applied. In the mutation, a new random vehicle and a new start time are assigned to one randomly selected trip. Crossover and mutation rates are defined as 70% and 1%, respectively, and termination condition is defined by 150 generations. The best found solution allowed decreasing the total idle time comparing with the original schedule.

In first series of optimisation experiments, simulation model with deterministic data is used. A series of optimisation experiments with population sizes of 200, 500, 1000 and 2000 are performed. Termination condition is set to occur when a large number of generations are generated without improvement of the best solution in the population. Optimisation results show that a solution which satisfies all constraints can be found. Acceptable results are obtained with the population size equal to 1000 chromosomes. Larger sizes of the population notably increase the search time of optimisation algorithm.

The caching of solutions gives a growth of optimisation algorithm's speed mostly after the time moment, when algorithm starts to converge toward an optimal solution, and the diversity of solutions becomes smaller, but the mutation becomes the main solution improvement operator.

In the second series of optimisation experiments, the optimisation of a model with stochastic moving times is performed. The found solutions are approximately the same in quality as solutions for the problem with deterministic data. The optimisation algorithm needs improvements, as many found solutions still do not satisfy part of soft constraints.

The proposed design of the genetic algorithm for the VSPTW has shown better results than general purpose optimisation tool OptQuest, which has stuck in the local optima and could not find any solutions that satisfied all constraints of the problem.

Fitness landscape analysis and optimisation in HeuristicLab

To perform a faster and more comprehensive analysis, the simulation model described in [32] was reimplemented as a plug-in of HeuristicLab [49] maintaining all logic of the described simulation model.

To enhance the quality of optimisation results, permutation encoding for the VSP solutions is introduced. The encoding is based on the Alba encoding [3] for the vehicle routing problem. A chromosome contains $m + n$ genes, where n is a number of vehicles and m is a number of trips. The genes that have values less or equal to *m* encode the trip number and values greater than *m* encode delimiters or vehicle designators, and define, that the next sequence of trips should be performed by the corresponding vehicle. If no time window constraints are defined for the first trip, it starts at midnight; otherwise it starts at the appropriate time to match the first customer's window. The next trip starts immediately after previous, unless its start time should be delayed to satisfy time windows of customers in the route of this trip. No times are encoded, and hence, the potential solution has no immediately encoded idle time and trips cannot overlap.

Currently there are no reference values to compare fitness landscape measures. Thus a grid of the landscape analysis experiments is created to compare values between different landscapes. The problem considered in [32] is called VSP_37 and its extension to 133 trips is called VSP_133. Additionally these two problems are analysed with different numbers of available vehicles. Three additional instances of the problems with unusual data VSP_s* and three instances with artificial data VSP_a* are created.

In this subsection, the fitness landscape experiments are divided into three large groups:

1. Comparison of different operators for the existing encoding;
2. Impact measurements of stochastic variables during simulation;
3. Comparison between existing and proposed encodings.

For the integer vector representation, fitness landscapes of two operators are analysed. Both operators uniformly select the gene at a random position and change the assigned vehicle for this trip. The single position replacement manipulator (*VSPManipulator*) changes the start time of the trip to a new uniformly distributed random number, but the single position shift manipulator (*VSPShiftManipulator*) shifts the start time with a uniformly distributed random number. Experiments are performed with 10 repetitions of 20 000 step long walk trajectories.

In random walk, values of autocorrelation function are slightly lower for the *VSPManipulator* than for other manipulators, because this operator allows big changes of the solutions. In up-down walk the situation is the opposite: replacement mutation has higher correlation than shift mutation, but the three artificial problems are different to the others (Fig. 5).

Neutral walks on the landscape reveal one instance (VSP_s2) with different information content, which means that this instance has higher neutrality, which was not observed in the correlation analysis due to the high autocorrelation for small lags.

The problems with equivalent input data, but with a different number of vehicles, have similar landscape measures in the experimental analysis. It can be concluded that for VSPs the local landscape structures are relatively irrelevant to the number of vehicles, while the main impact on them has a number and variety of trips.

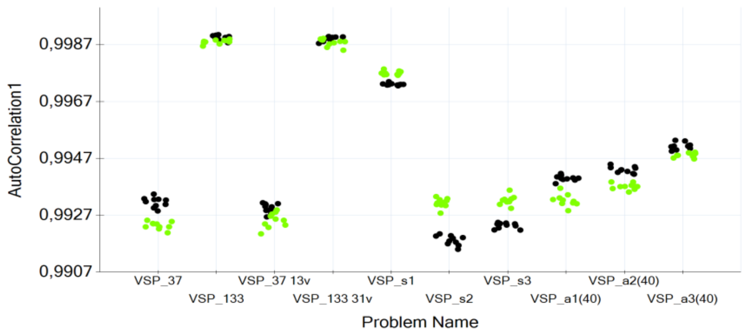


Fig. 5. Autocorrelation obtained in up-down walks (black dots – Replacement Mutator; green – Shift Mutator)

The developed plug-in was supplemented with additional logic to estimate the affect of simulation model's stochastic variables on the landscape measures. In this logic, all vehicle movement times are shifted by a random number that has symmetric triangle distribution in the interval [-20; +20] minutes. Ten experiments for each type of evaluator are performed with three different types of walks, each with 20 000 steps and with application of *MultiVSPManipulator* mutation operator, which randomly applies *VSPManipulator* or *VSPShiftManipulator*.

The autocorrelation value ρ(1) is lower for landscapes of noisy problems with the exceptions of the real-life instances in random walks, where the difference between autocorrelation values is negligible. At the same time this difference is higher for problems with artificial data, where the sequence of trips could be very compact. The addition of similar noise has different impact on different problems, which can be measured by $H(0)$ in random walks. Similar results are obtained for the partial information content $M(0)$ which values are higher for landscapes with noise and vice versa, density-basin information $h(0)$ is higher for problems with deterministic moving times, as the landscapes of such problems have larger basins of attraction due to a lower modality.

To find in what way the number of simulation model replications affects the landscape analysis measures, additional experiments were performed: one series with the deterministic problem VSP_37, and three stochastic series with 1, 5 and 10 replications of the stochastic model of VSP_37. In each series, 20 random walks with 10 000 steps were performed. No significant difference of correlation length and autocorrelation values between different numbers of replications was found. The information content value is higher for the problem with additional noise, especially when only one replication is used. The partial information content and density basin information show identical behaviour. The information measures are sensitive to the noise in the fitness function, and higher number of replication reduces the impact of the noise.

By comparing the problems with different number of trips, the following results were found: in all types of landscape walks, autocorrelation ρ(1) for deterministic problem is higher than for stochastic, and this difference is higher for problems with the higher number of trips, and not so obvious for smaller problems.

To compare the fitness landscape analysis measures between different VSP representations, ten experiments for each encoding are performed with three different types of walks, each with 20 000 steps. The value of the autocorrelation function in random and up-down walks is lower for the permutation encoding for all problem instances. This means that landscapes of this encoding should be more rugged. There is an exception in neutral walk, where VSP_s* problems have higher autocorrelation for landscapes of permutation representation, which means that these problems have large neutrality in this encoding. The information content values obtained in random walk are similar between all problems in any encoding, except for the VSP_s1 and VSP_s2 problems in permutation representation, where it is very low. In up-down walks, values of the information content for all instances are higher for the permutation encoding. In the neutral walk, the information content is small for all problems in the permutation encoding.

Evolution Strategy (ES), Simulated Annealing (SA) and Genetic Algorithm were applied in the comparison of VSP optimisation results. For integer encoding, both ES and SA algorithms are fast and highly successful, and it is possible to find solutions with better quality with ES. GA finds even better solutions, but requires a higher number of evaluations. Optimisation algorithms find good solutions faster with application of the permutation encoding, thus it is more effective in optimisation of the VSP. Even though the search space for the permutation encoding is more complex and rugged, nevertheless, due to its smaller size it is more effective in the search of the global optimum. In both encodings ES is more effective than GA.

Optimisation experiments with similar instances and parameters show that there are important relations between the values of the fitness landscape analysis and optimisation performance. To obtain these relations, GA with population size of 100 individuals and 500 generations was applied in one series of optimisation experiments, and (20+100)-ES with a crossover and 1 000 generations in the second. By comparing optimisation results for different mutation operators, it can be seen that the statistical analysis can predict the performance of the mutation operators. For the GA, the shift operator is better for problems VSP_s1 and VSP_s2. These instances also have higher autocorrelation for the shift mutation operator (Fig. 5, 6). The same dependency was also found for the ES algorithm.

In ES optimisation experiments the highest difference between stochastic and deterministic instances is for the problems with higher information content, but for the GA this dependency is not so significant.

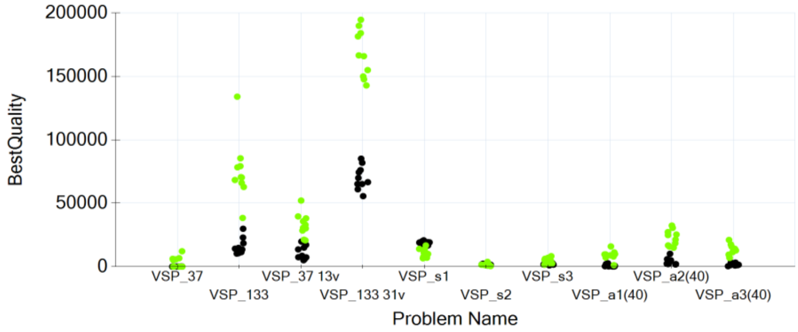


Fig. 6. Quality of best found solutions with ES (black dots – Replacement Mutator; green – Shift Mutator)

In the experiments with different encodings, the most interesting results were obtained for the GA. Here, for a large part of small instances, performance of permutation encoding is higher, and for the largest problem (VSP_133) it is significantly lower (Fig. 7). A possible explanation of this behaviour is that gain of the permutation encoding is due to its reduced search space. As it was noted in the ruggedness analysis, the landscapes in the permutation encoding should be harder to search in. Due to the large search space of the VSP_133 problem, the factor of landscape ruggedness dominates the reduction of the size of search space. At the same time in ES optimisation, permutation encoding has shown very high performance, and the global optimum was found in almost all instances.

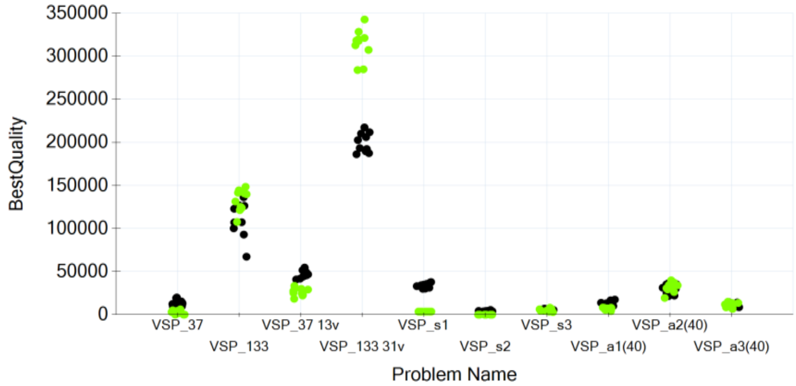


Fig. 7. Fitness of best found solutions with GA for different encodings (green dots – permutation, black - integer)

The recommendations for metaheuristic optimisation of the VSP are defined in the conclusions of this chapter. Optimisation using the ES is the best choice for the solution of the vehicle scheduling problem with time windows. If the GA is selected as the optimisation algorithm, permutation encoding has to be chosen unless the problem contains more than 100 trips. In the integer vector encoding, selection of the appropriate optimisation operator is based on the landscape analysis measures: an operator, which has the highest autocorrelation value in the up-down walk, should be selected.

4. Application in Product Delivery Planning

Two combined optimisation tasks of the integrated delivery planning are solved in this section. These are the tasks that are solved on the last step of the delivery planning integrated methodology proposed in [29, 30]. The list of shops and the amount of goods that should be delivered in each shop are the input data of these two tasks.

Completed task of this level is the determination of the best routes and schedule for the vehicles to deliver goods from distribution centre (DC) to the shops. Optimal distribution of routes and vehicles should minimise a number of used vehicles and total delivery distances, with minimisation of vehicle idle times. The route and schedule plan must fulfil the constraints, such as capacities of vehicles, time windows and warehouse capabilities. Two similar, but not identical problem statements and existing tools are used in the solution of the combined vehicle routing and scheduling task with time windows.

For the solution of this combined task, two optimisation problems are solved sequentially by application of the adjusted optimisation tools. First task is solved as a classical vehicle routing problem with time windows (VRPTW), which aims optimisation of vehicle routes. For the second task a route scheduling problem statement is defined that aims optimisation of a schedule of predefined routes.

Vehicle routing problem statement

The VRP is a multiple travelling salesman problem, where a demand is associated with each city and the salesmen are interpreted as vehicles each having the same capacity. The sum of demands on a route cannot exceed the capacity of the vehicle assigned to this route. It is required to minimise the sum of distances of the routes. In the capacitated VRP the demand may be constraining routes. If a time slot, in which customers have to be visited, is added to each customer, then the VRP with time windows (VRPTW) is obtained. In addition to the capacity constraint, a vehicle has to visit a customer within a certain time interval given by a ready time and a due time. It is allowed for a vehicle to arrive before the ready time, but it is forbidden to arrive after the due time. Often the number of customers combined with the complexity of real-life data does not permit solving a problem exactly. In these situations it is commendable to apply approximation algorithms, heuristics or metaheuristics [2].

The mathematical formulation of the general VRPTW is based on the model defined in [7]. VRPTW is given by a fleet of homogeneous vehicles V, a set of customers C and a directed graph G. The graph consists of $|C|+2$ vertices, whereby the customers are denoted as 1, 2,..., n and the depot is represented by the vertices 0 and $n+1$. The set of vertices is denoted as N; the set of arcs A represents connections between customers and between the depot and customers, where no arc terminates in vertex 0 and no arc starts from vertex n+1. With each arc (i, j), where $i \neq j$, a cost c_{ij} and a time t_{ij} are associated, which may include service time at the customer i. Each vehicle j has a capacity q_j and each customer i has a demand d_i. Each customer i has a time window [a_i, b_i]; a vehicle can arrive before a_i, but it must arrive before b_i. In the general description, the depot also has a time window, which is the scheduling horizon of the problem: vehicles may not leave the depot before a_0 and must return back before or at time b_{n+1}. It is postulated that q, a_i, b_i, d_i and c_{ij} are non-negative integers, while the t_{ij} values are assumed to be positive integers. This model contains two sets of decision variables, namely x and s. For each arc (i, j), where $i \neq j$, $i \neq n+1$, $j \neq 0$, and each vehicle k: $x_{ijk} = 0$ if vehicle k does not drive from a vertex i to a vertex j, and $x_{ijk} = 1$ if vehicle k drives from the vertex i to the vertex j. The decision variable s_{ik} is defined for each vertex i and each vehicle k denoting the time, when the vehicle k starts a service at the customer i. It is assumed, that $a_0 = 0$ and therefore $s_{0k} = 0$ for all k.

The goal is to design a set of routes with minimal cost, one for each vehicle, such that: each customer is serviced exactly once; every route originates at vertex 0 and ends at vertex n+1; the time windows and capacity constraints are complied with.

The goal function of the VRPTW is stated as follows:

$$\sum_{k\in V}\sum_{i\in N}\sum_{j\in N} c_{ij}x_{ijk} \rightarrow \min \qquad (32)$$

A number of constraints are described in [7]. Constraints state that each customer is visited exactly once and that no vehicle is overloaded; ensure that each vehicle sequentially visits all points in the route; that a vehicle k cannot arrive at j before $s_{ik} + t_{ij}$ if it is travelling from i to j; and ensure that the time windows are adhered [2].

Vehicle routing experiments

To perform the optimisation experiments of the VRPTW, its input data based on the data of the case study was prepared. In the case study the product delivery plan of one day is specified for the groups of goods that are delivered jointly in the same vehicle. For each group of goods individual series of optimisation experiments were performed, based on the union of the goods in the deliveries. For each of these groups separate “virtual” shops with same coordinates, but different demand are defined for the chosen day.

All experiments were performed with application of the Island Offspring Selection Genetic Algorithm (IOSGA) [2], which is a special type of genetic algorithm that combines features of coarse-grained parallel (island) GA and GA with offspring selection. Choice of algorithm is based on the high optimisation pressure and adjustment abilities of the IOSGA. Experiments were performed with HeuristicLab optimisation framework

[49], with application of its IOSGA and VRP plug-ins. In all experiments of this section following parameters of the optimisation algorithm are defined: proportional selection operator and *MultiVRPMutator* mutation operator from HeuristicLab operators with 5% probability; population size 200 with 1 elite solution; population divided in 5 islands and migration of best solutions between islands is performed with the periodicity of 20 generations. Selection of the crossover operator is described below. Maximal offspring selection pressure of 200 was defined as termination condition.

Following input data of the routing problem was prepared for the VRP plug-in: capacity of the vehicles in roll-containers; coordinates of each shop and distribution centre; the daily demand of shops; the due and ready times of the shops in minutes since midnight; service time of each shop in minutes; a number of available vehicles. Coordinates of shops were defined so that Euclidean distance between two points was approximately equal to the driving time between these points in minutes.

A set of optimisation experiments was performed to find the crossover operator which provides most qualitative solutions. The GVR [36], edge recombination (ERX) [51] and maximal preservative (MPX) [34] crossovers for solutions encoded in Alba encoding [3] were compared in these experiments. Application of the ERX crossover provided the best results in terms of the total distance, but application of GVR crossover allowed better satisfaction of the capacity constraints. A GVR crossover was selected as best for the VRPTWs of the case study, as it works with an unlimited number of vehicles, but provides best results in terms of keeping routes not overloaded. To minimise a number of required vehicles later, the vehicle route scheduling problem is introduced above.

The visualisation of the best found solution in one of the optimisation experiments for the deliveries of “Dry” goods of one day is shown in Fig. 8. A blue circle here defines the distribution centre, black points define locations of the shops that were served within the time windows and yellow points are the shops, which will be the first to be served in a route. The capacity of all vehicles was set to 30 roll-containers. In best found solutions all customers outside of central city have to be served with long routes including 3 to 4 shops. Corresponding delivery routes that include shops located in the city are shown in the inset of Fig. 8. These routes are only 1 to 3 shops long. The number of shops that are served exclusively is higher, because of the higher demand of the city stores. GA experiments have stopped on the 30-50 generation, when the selection pressure has achieved its limit. The best found solution of the considered instance requires 34 vehicles to serve all shops with each vehicle having only one route.

In similar way optimisation experiments for the “POD Cooler” and “F&V” good types were performed. First runs were performed with vehicle capacity equal to 30 containers, but experimental results show that vehicles with this capacity are too small for the specified demand of shops. Second series of experiments were performed with larger vehicle capacity, defined as 60 containers.

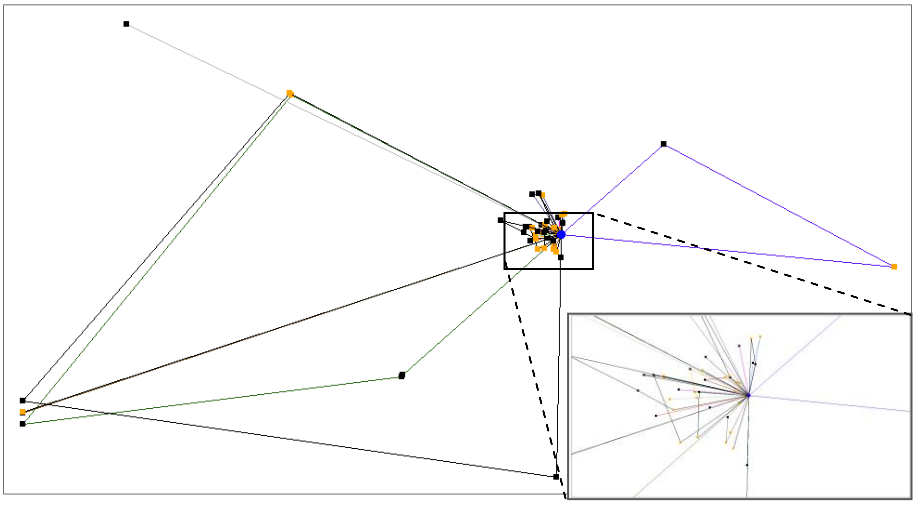


Fig. 8. Routes for Dry goods, for vehicles with capacity 30

Number of shops in the optimal routes is limited due to the small capacity of the vehicles, and not because of short time windows, which can be seen in the central part of the visualisation (Fig. 8).

Vehicle route scheduling problem statement

It is assumed in the definition of the classical VRPTW, that any vehicle may perform only one route in the planning horizon. In the investigated business case, all routes are shortened by the capacity of vehicles, which leads to the ineffective solutions of the vehicle routing problem. To overcome these obstacles, the route scheduling problem is introduced. It can be formulated on a basis of the Vehicle Scheduling Problem with Time Windows (VSPTW) and solved with methods and tools developed in the Section 3. In the formulated problem, the routes correspond to the trips in the VSPTW task. Vehicles may perform any fair number of routes during the day. As far as the final solution of the VRPTW task should be feasible for the capacity and time window constraints, it can be optimised by combining and compacting routes to increase a vehicle utilisation. Application of the vehicle scheduling for the solution of vehicle routing problem allows reducing a number of required vehicles.

The problem statement described in Section 3 and [28] has been modified. For each group of goods routing is performed separately. Time windows and service times are introduced for all customers. Input data of the vehicle routing problem partly is used as input data of the scheduling problem. The sequence of shops in trips in the proposed statement is defined as a route, and the moving times in a trip are interpreted as transportation times. A vehicle capacity is not involved in the statement.

A formal statement of the route scheduling problem includes a set of customers (shops) N and a list of routes R, which are obtained from the VRPTW solution. Each route defines a sequence of visited customers. Statement includes a set of transportation times, which define vehicle moving times t_{ij} between the route's sequential points i and j; a set of vehicles V and an estimated number of vehicles $|V|$. For each shop i, its time window is defined as ready time a_i and due time b_i. Also for the customer i a service time z_i in minutes is defined.

Decision variables are ones introduced in the routing model, i.e., sets x and s, except that $x_{ijk} = 1$ states that for vehicle k route j will be the next after route i. Two types of soft constraints are introduced in the problem: 1) time window constraint; 2) overtime constraint. Time window constraint in the problem is monitored by a number of times N_{ad}, when a vehicle leaves a customer after the due time. The problem statement allows vehicles to arrive to a customer before the ready time, but the time, while the vehicle will wait for the ready time is counted as the idle time. Overtime constraint in the solution candidate is defined as a number of vehicles N_{ot} with required total delivery time more than 24 hours. Additional constraints are introduced in the problem statement to assure the integrity of the model and the provision of schedule simulations.

A fitness function f of the route scheduling problem summarizes all idle times, which occur due to fitting deliveries into the time windows, and a number of constraint violations multiplied by penalty values:

$$f = \sum_{k \in V} l_k + p_{ad} N_{ad} + p_{ot} N_{ot} \to \min, \quad (33)$$

where l_k is the total idle time of a vehicle k; V is a set of available vehicles; N_{ad} is a number of vehicles, which leave customer after due time; N_{ot} is a number of vehicles, which are scheduled to work with overtime; p_{ad} and p_{ot} are the penalty values for late deliveries and vehicle overtimes, correspondingly, and are assumed to be significantly greater than 1.

As the route scheduling problem is derived from the above described VSPTW, an application of the evolution strategy algorithm is proposed for its optimisation, because this method has demonstrated in experimental analysis (see Section 3) the best results in optimisation of the VSPTW.

Vehicle scheduling optimisation experiments

Two types of the vehicle scheduling experiments with the input data of the investigated case study are performed in the thesis. The first are based on the VSPTW problem statement described in Section 3. The second series are based on the route scheduling problem statement defined in Section 4.

VSPTW experiments are performed using the HeuristicLab framework and the developed plug-in of vehicle schedule problem. Optimisation is performed with the (20+100)-ES. Termination condition is defined as 1000 generations, and mutation operators are selected accordingly to the representation. The VSPTW optimisation experiments were performed both with integer and permutation encodings.

The Gantt chart of the best found solution, which is visualised in the main window of the developed VSP plug-in, for an optimisation run with permutation encoding is shown in Figure 9. For the integer encoded solutions idle times are big, although all constraints in solutions are satisfied. Results for the permutation encoding are better in terms of quality of best found solutions. All trips in this encoding are combined in a compact manner, thus no idle times are left at all. A number of vehicles in the demonstrated solution (Fig. 9) can be reduced.

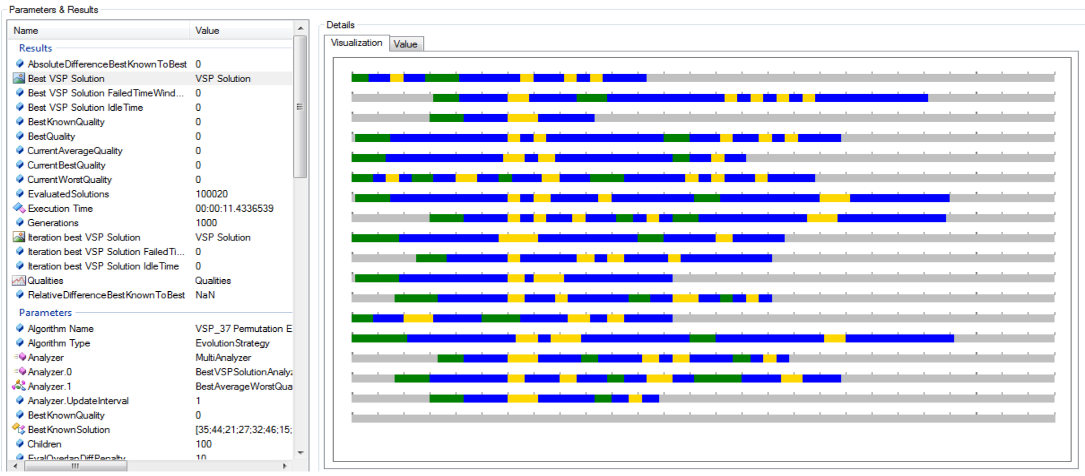


Fig. 9. VSPTW solution with permutation encoding

To resolve the vehicle route scheduling problem, a plug-in in HeuristicLab optimisation framework is developed. In the plug-in, fitness function (33) evaluator simulates a schedule of a solution candidate and identifies time windows mismatches, evaluates idle times and the total usage time for each vehicle. A permutation encoding of the VSPTW is applied for the route scheduling, but the trips here are represented by the routes. Input data of the route scheduling plug-in are vectors of the ready, due and service times of shops, lists of routes and moving times between route points and a number of available vehicles. All times are given in minutes. Ready and due times of time windows are given in minutes since midnight.

Several series of optimisation experiments were performed to determine a suitable algorithm for the route scheduling. Following algorithms were examined: Evolution Strategy, Genetic Algorithm, Island Genetic Algorithm with 5 islands (IGA) and Offspring Selection Genetic Algorithm (OSGA) [2]. Maximal preservative crossover [34] and insertion manipulator were defined as genetic operators for all algorithms. For genetic algorithms the proportional selection was applied. To determine a suitable algorithm, numbers of solution evaluations performed to obtain candidate solutions of the equal fitness were compared on instances, with a low number of vehicles. The results of comparisons show potentially small effectiveness of the crossover operator against a mutation operator. The ES was chosen as most suitable, for its ability to provide globally optimal results of the vehicle route scheduling with fewer evaluations. As the route scheduling problem is derived from the VSPTW, the optimisation algorithms show the same difference in the performance.

A sample experiment based on one day plan and specific demand data for 53 shops is described. The best found solution obtained by the IOSGA for the VRPTW defines 34 routes (see Figure 10). Most of the vehicles in the solution have very short routes due to a small vehicle capacity. Herewith, it is possible to combine these routes due to the long time windows. The ES (20+100) algorithm was applied for the route scheduling problem which input data is based on the considered VRPTW solution. As a result, the globally optimal scheduling solutions were found with all constraints satisfied if at least 6 vehicles are available. The correspondent Gantt chart is shown in Figure 10. Green lines in the timelines correspond to the loading process in the DC, blue ones to the transportation, and yellow lines correspond to the vehicle unloading at the stores.

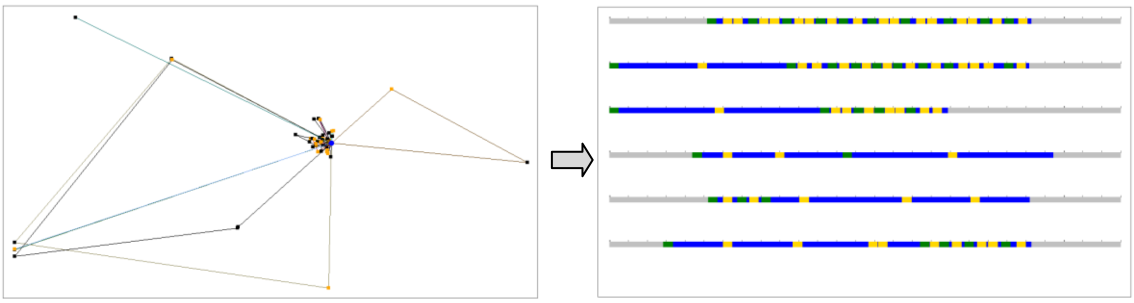


Fig. 10. VRPTW and scheduling solution of case instance

The proposed vehicle scheduling method that complements vehicle routing ensures effective route and schedule solutions for a short-term delivery planning. This method can be applied in the vehicle routing and scheduling tasks, where routes are very short in comparison with a planning horizon.

RESULTS AND CONCLUSIONS OF THE THESIS

The aim of the doctoral thesis was to develop the methods and algorithms for the simulation-based fitness landscape analysis and optimisation of complex systems.

The results and conclusions of the thesis are as follows:

1. State-of-the-art analysis in the simulation-based optimisation for NP-hard problems allowed for selection of metaheuristic algorithms as the most efficient for the optimisation of NP-hard

problems. Review of the numerical simulation-based optimisation methods, review of formal definitions of fitness landscape and definitions of its structures, with the review of information and statistical methods and measures of fitness landscape analysis allowed development of the formal scheme for the simulation-based optimisation, enhanced with the fitness landscape analysis.

2. The developed formal scheme of the simulation-based optimisation with fitness landscape analysis allowed extension of the fitness landscape analysis methods on simulation-based optimisation tasks and identifying requirements for methods, algorithms and software tool prototype for simulation-based fitness landscape analysis and optimisation.
3. Experimental fitness landscape analysis of benchmark landscapes with different operators and representations allowed finding the relations and dependencies between structural features of benchmark fitness landscapes, their information and statistical measures and behaviour of optimisation algorithm on these landscapes. Experimental analysis of noisy fitness functions allowed for determination of additional requirements for simulation-based fitness landscape analysis.
4. The developed algorithms for the simulation-based fitness landscape analysis procedure allowed implementation of a software tool prototype for fitness landscape analysis. Application of the developed tool provided analysis of the vehicle scheduling problem with time windows in simulation-based optimisation.
5. The comprehensive experimental fitness landscape analysis of the vehicle scheduling problem with time windows in the optimisation framework allowed determination of problem specific properties and internal characteristics of problem fitness landscape. This analysis allowed determination of impact of problem specific properties on the structure of a fitness landscape and performance of optimisation algorithm. This, in turn, provided development of recommendations for best optimisation scenarios for the vehicle scheduling problem with time windows.
6. The developed methods and algorithms were applied in the solution of delivery planning operational level optimisation tasks, which allowed improving the overall solutions of vehicle routing and scheduling problem with time windows. Application of the adjusted metaheuristic algorithms for optimisation of defined routing and scheduling tasks allowed decrease of a number of required vehicles and provided more effective vehicle utilisation.

MAIN INFORMATION SOURCES

1. Ackley D. An empirical study of bit vector function optimization. // Genetic Algorithms and Simulated Annealing. - 1987. - pp. 170-215.
2. Affenzeller M., Winkler S., Wagner S., Beham A. Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications. - Chapman and Hall/CRC, 2009. - 379p.
3. Alba E., Dorronsoro B. Solving the vehicle routing problem by using cellular genetic algorithms. // Lecture Notes in Computer Science Vol. 3004. - Springer, 2004. - pp. 11-20.
4. Beerenwinkel N., Pachter L, Sturmfels B. Epistasis and Shapes of Fitness Landscapes // Statistica Sinica Vol. 17, No. 4. - 2007. - pp. 1317-1342.
5. Biles, W.E. A Gradient Regression Search procedure for Simulation Experimentation // Proceedings of the 1974 Winter Simulation Conference. - 1974. - pp. 491-497.
6. Collard P., Verel S., Clergue M. Local Search Heuristics: Fitness Cloud versus Fitness Landscape // 16th European Conference on Artificial Intelligence. - 2004. - pp. 973-974.
7. Cordeau F., Desaulniers G., Desrosiers J., Solomon M.M., Soumis F. VRP with Time Windows // Toth P. and Vigo D. (Eds.), The vehicle routing problem. - Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2001. - pp. 157-193.
8. Czech Z.J. Statistical measures of a fitness landscape for the vehicle routing problem. // IEEE International Symposium on Parallel and Distributed Processing. - 2008. - pp. 1-8.
9. De Jong K. D. An analysis of the behavior of a class of genetic adaptive systems. PhD thesis. - Ann Arbor, USA: Department of Computer and Communication Sciences, University of Michigan, 1975. - 256p.
10. Dreo J., Petrowski A., Siarry P., Taillard E. Metaheuristics for Hard Optimization. Methods and Case Studies. - Springer-Verlag, Berlin Heidelberg, 2006. - 369 p.
11. Eiiyi D.T., Ornek A., Karakutuk S.S. A vehicle scheduling problem with fixed trips and time limitations // International Journal of Production Economics, Vol. 117, No.1. - 2008. - pp. 150-161.
12. Glover, F.; Kochenberger, G.A. Handbook of metaheuristics. International Series in Operations Research & Management Science. - Springer, 2003. - 570p.
13. Glover F. Tabu Search - Part I // INFORMS Journal on Computing, Vol. 1, No. 3. - 1989. - pp. 190-206.
14. Goldberg, D. E. Genetic Algorithms in Search, Optimization and Machine Learning. - Addison-Wesley Professional, 1989. - 432 p.
15. Gosavi A. Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning. - Kluwer Academic Publishers, 2003. - 554 p.
16. He J., Yao X., Zhang Q. To Understand Fitness Landscapes in Continuous Space by Using Drift Analysis // Congress on Evolutionary Computation, CEC2004, Vol. 2. - 2004 - pp. 1248-1253.
17. Holland J.H. Adaptation in Natural and Artificial Systems. - University of Michigan Press, 1975. - 183p.
18. Hordijk W. A Measure of Landscapes // Evolutionary computation, Vol. 4. - 1996. - pp. 335-360.
19. Horn J., Goldberg D.E. Genetic Algorithm Difficulty and the Modality of Fitness Landscapes // Foundations of Genetic Algorithms, Vol. 3. - Morgan Kaufman,

1994. - pp. 243-269.
20. Jones T. Evolutionary Algorithms, Fitness Landscapes and Search. PhD Thesis. - Albuquerque: The University of New Mexico, 1995. - 224p.
21. Jones T., Forrest S. Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms // Proceedings of the Sixth International Conference on Genetic Algorithms. – San Francisco, CA: 1997. - pp.184-192.
22. Kauffman S. Adaptation on rugged fitness landscapes // Lectures in the Science of Complexity. - Addison-Wesley, 1989. - pp. 527-618.
23. Kinnear K. E. Fitness Landscapes and Difficulty in Genetic Programming // Proceedings of the 1994 IEEE World Conference on Computational Intelligence. - Orlando, Florida, USA: 1994. - pp. 142-147.
24. Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P. Optimization by Simulated Annealing // Science, Vol. 220, No. 4598. - 1983. - pp. 671-680.
25. Knuth D. The Art of Computer Programming. Vol. 2: Seminumerical Algorithms. - Reading, Massachusetts, USA: Addison-Wesley, 1997. - 762p.
26. Langdon W.B., Poli R. Foundations of Genetic Programming. - Berlin, Heidelberg: Springer-Vorlag, 2002. - 260p.
27. Manderick B., Weger M., Spiessens P. The Genetic Algorithm and Structure of the Fitness Landscape // Proceedings of the Fourth International Conference on Genetic Algorithms. - 1991. - pp. 143-150.
28. Merkuryeva G., Bolshakov V. Simulation-based fitness landscape analysis and optimisation for vehicle scheduling problem // EUROCAST 2011, Part I, LNCS 6927. - Springer, 2011. - pp. 280-286.
29. Merkuryeva G., Bolshakov V. Simulation Optimisation and Monitoring in Tactical and Operational Planning of Deliveries // Proceedings of the European Modeling and Simulation Symposium. - 2012. - pp 226-231.
30. Merkuryeva G., Bolshakov V., Kornevs M. An Integrated Approach to Product Delivery Planning and Scheduling // Scientific Journal of RTU. 5. series., Computer Science. Vol. 49. - Riga: RTU Publishing House, 2011. - pp. 97-103.
31. Merkuryeva G., Bolshakovs V. Comparative Analysis of Statistical and Information Measures for Benchmark Fitness Landscapes // Proceedings of the UkSIM Fourth European Modelling Symposium on Computer Modelling and Simulation. - 2010. - pp. 96-101.
32. Merkuryeva G., Bolshakovs V. Vehicle Schedule Simulation with AnyLogic // 12th International Conference on Computer Modelling and Simulation. - Cambridge, 2010. - pp. 169-174.
33. Merz P., Freisleben B. Fitness Landscape Analysis and Memetic Algorithms for the Quadratic Assignment Problem // IEEE Transactions on Evolutionary Computation, Vol. 4, Issue 4. - 2000. - pp. 337-352.
34. Muhlenbein H. Evolution in time and space - the parallel genetic algorithm // Foundations of Genetic Algorithms. - Morgan Kaufmann, 1991. - pp. 316-337.
35. Nagamochi H., Ohnishi T. Approximating a vehicle scheduling problem with time windows and handling times // Theoretical Computer Science, Vol. 393, Issue 1-3. - 2008. - pp. 133-146.
36. Pereira, F.B., Tavares, J., Machado, P., Costa E. GVR: A New Genetic Representation for the Vehicle Routing Problem // Proceedings of the 13th Irish International Conference on Artificial Intelligence and Cognitive Science (AICS '02). - Limerick, Ireland, 2002. - pp. 95-102.
37. Pitzer, E., Affenzeller, M. A Comprehensive Survey on Fitness Landscape Analysis // Recent Advances in Intelligent Engineering Systems. - Springer, 2011.

- pp. 167–196.
38. Pitzer E., Vonolfen S., Beham A., Affenzeller M., Bolshakov V., Merkuryeva G. Structural Analysis of Vehicle Routing Problems using General Fitness Landscape Analysis and Problem Specific Measures // 1st Australian Conference on the Application of Systems Engineering (ACASE'12). - Australia, Sydney: 2012. - pp. 36-38.
39. Reeves C.R., Rowe J.E. Genetic Algorithms - Principles and Perspectives. A Guide to GA Theory. – Springer, 2002. - 344p.
40. Reidys C.M, Stadler, P.F. Neutrality in Fitness Landscapes // Appl. Math. & Comput., Vol. 117. - 1998. - pp. 321-350.
41. Schwefel H.P. Evolution and Optimum Seeking. - Wiley-Interscience, 1995. - 465p.
42. Sipser M. Introduction to the Theory of Computation. Second Edition. - Thomson Course Technology, 2006. - 431p.
43. Smith, T., Husbands, P., Layzell, P., O'Shea, M. Fitness landscapes and evolvability // Evol. Comput., Vol. 10, No. 1. - 2002. - pp. 1-34.
44. Stadler P.F. Fitness Landscapes // Appl. Math. & Comput. Vol. 117. - 2002. - pp. 187-207.
45. Stadler P.F. Towards a Theory of Landscapes // Complex Systems and Binary Networks. Lecture Notes in Physics, Vol. 461. - 1995. - pp. 78-163.
46. Teo J.T.W. Pareto Multi-Objective Evolution of Legged Embodied Organisms. PhD Thesis. - Sydney: University of New South Wales, 2003. - 281p.
47. Vanneschi L., Tomassini M., Collard P., Verel S. Negative Slope Coefficient: A Measure to Characterize Genetic Programming Fitness Landscapes // Lecture Notes in Computer Science, Vol. 3905 - Springer, 2006. - pp. 178-189.
48. Vassilev V.K., Fogarty T.C., Miller J.F. Information Characteristics and the Structure of Landscapes. // Evolutionary Computation, Vol. 8, Issue 1. - Cambridge, MA, USA: MIT Press, 2000. - pp. 31-60.
49. Wagner S. Heuristic Optimization Software Systems - Modeling of Heuristic Optimization Algorithms in the HeuristicLab Software Environment. PhD Thesis. - Linz, Austria: Institute for Formal Models and Verification, Johannes Kepler University, 2009.
50. Weinberger. E. Correlated and Uncorrelated Fitness Landscapes and How to Tell the Difference // Biological Cybernetics, Vol. 63, Issue 5. - Springer-Verlag, 1990. - pp. 325-336.
51. Whitley D.L., Starkweather T., Fuquay D. Scheduling Problems and Traveling Salesmen: The Genetic Edge Recombination Operator // Proceedings of the 3rd International Conference on Genetic Algorithms, J. David Schaffer (Ed.). - San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989. - pp. 133-140.
52. Wright S. The roles of mutation, inbreeding, crossbreeding and selection in evolution. // Proceedings of the Sixth International Congress on Genetics, Vol. 1. - 1932. - pp. 356-366.
53. Растригин Л.А. Системы экстремального управления // Теоретические основы технической кибернетики. - М.: Наука, 1974. - 632с.