Simulation in Produktion
und Logistik 2019
Matthias Putz & Andreas Schlegel (Hrsg.)
Wissenschaftliche Scripten, Auerbach 2019

An Approach for Reducing the Search Space for Simheuristics Applications in Logistics Networks in Trading

Ein Ansatz zur Reduzierung des Suchraums für Simheuristik-Anwendungen in Logistiknetzen im Großhandel

Markus Rabe, Majsa Ammouriova, Dominik Schmitt, Jorge Chicaiza-Vaca,
TU Dortmund, Dortmund (Germany), {markus.rabe, majsa.ammouriova,
dominik.schmitt, jorge.chicaiza}@tu-dortmund.de

Abstract: For the purpose of improving the performance of logistics networks, so-called logistics assistance systems (LASs) are used. A LAS supports decision makers in taking actions, such as increasing the stock level of a stock keeping unit. A LAS can be combined with simulation to evaluate possible actions. With the size of the logistics network increasing, the number of actions to be evaluated raises and, therefore, the response time of the LAS. In this paper, the authors present two novel approaches for increasing efficiency and effectiveness of a given LAS by defining redundant and interchangeable actions. The results show that the number of simulation runs has been reduced by utilising the presented approaches; thus, the LAS's performance could be improved.

1 Introduction

Logistics networks in trading are complex systems. A decision maker (DM) needs to understand the internal structure of these networks to conduct promising decisions. To assist DMs, decision support systems (DSSs) for logistics networks, also called logistics assistance systems (LASs), have been developed (Liebler et al. 2013).

The authors have developed such a LAS for logistics networks in materials trading (Rabe et al. 2017; 2018a; 2018b). A simplified architecture of the LAS is presented in Figure 1. The authors' LAS determines promising actions and suggests them to the DM in order to help coping with the complexity of the logistics network. Such an action may change the stock level of a stock-keeping unit (SKU) at a site or alter the logistics network's transportation structure.

Promising actions are identified by the heuristic unit (HU). For this purpose, the HU explores the search space, consisting of all possible actions for a given network's state. Actions are used to form an action plan, comprising selected actions that will be applied sequentially to the simulation model. The performance of the logistics network is evaluated by discrete event simulation (DES). The simulation results are provided to the HU, which synthesises and applies further action plans until a specific termination condition is met, e.g., reaching a predefined number of iterations.

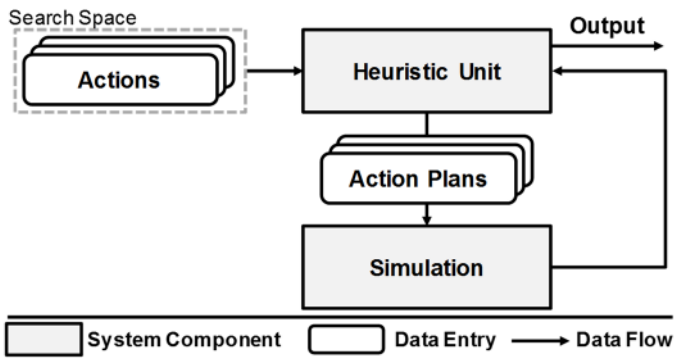

Figure 1: *Simplified architecture of the author's LAS based on Rabe et al. (2017)*

However, as the logistics network increases in size, the response time of the LAS increases as well, under the additional computational load. The additional burden is due to an increased search space and, therefore, an increased response time of the HU. In addition and even more important, the computational time for each simulation run increases, as the logistics network increases, as well.

In order to address this issue, the authors search for approaches to reduce the system's overall response time. For instance, domain-specific information has been used in order to guide the HU's search for promising actions (Rabe et al. 2018a). In addition, the authors investigated the possibility of reducing the number of actions by grouping similar actions together, leading to a shorter response time of the LAS due to faster convergence (Rabe et al. 2018b).

In this paper, the authors present a novel approach for reducing the number of simulation runs and, thus, the LAS's overall response time, by using a single simulation run as an evaluation for a number of similar action plans. Therefore, the authors present a set of rules for specifying and identifying those similar action plans.

The paper is organised as follows: First, an overview of related work is given. Afterwards, a detailed description of the author's approach for reducing the number of simulation runs, including corresponding experiments and results, is provided. The paper closes with an outlook.

2 Related Work

This section presents logistics assistance systems and heuristics in the context of logistics. Furthermore, pattern recognition is presented.

2.1 Logistics Assistance Systems

Decision support system (DSS) is a widely-used term. A DSS is defined by Turban (2007) as "a system intended to support managerial decision makers in semistructured and unstructured decision situations". In general, a DSS receives data from a real system, elaborates them and finally suggests decisions to a DM.

A special support system for logistics managers is a LAS, such as the one defined by Blutner et al. (2009), who identify critical logistics situations and offers possibilities to evaluate the consequences of possible decision alternatives objectively. Throughout the literature, the terms LAS and DSS in the domain of logistics are mostly used synonymously. In this work, the term LAS is used.

Logistics assistance systems can use different simulation approaches. Simulation can be utilised to support managers with more insight and assist them in the management of logistics networks. One of the most widely-used simulation approaches is DES with hundreds of applications (Tako and Robinson 2012). In DES models, the system's behaviour remains in a specific state for a discrete period of time. The behaviour of the system changes from state to state over time, but the system remains in a specific state for a specific duration of time.

Regarding further research, Liebler et al. (2013) presented a simulation-based approach for gaining insight into global supply networks and explain its use for LAS. Rabe et al. (2018a) extended the development of a LAS, which uses DES to evaluate the consequences of possible actions in trading networks.

2.2 Heuristics in the Context of Logistics Networks

Evolutionary algorithms (EAs) are one of many metaheuristic optimisation techniques that have been widely adopted by researchers for solving production and logistics problems (Ławrynowicz 2011). EAs stand for a class of stochastic optimisation methods inspired by the "survival of the fittest". They work iteratively on a population of candidate solutions of a given problem. In the population, individuals represent candidate solutions. The population is evolved over generations to produce better solutions for the given problem. The evolution of the population from one generation to the next is usually achieved through the use of three operators: (i) selection in which individuals are chosen from a population for later breeding; (ii) mutation in which diversity from one generation of a population to the next is maintained; and (iii) crossover in which information of two parents are combined to generate new offspring (Ławrynowicz 2011).

Juan and Rabe (2013) described the fundamental concept behind the simheuristics term as a particular simulation for an optimisation approach. It is oriented efficiently to tackle an optimisation problem involving stochastic components. They reviewed some practical applications in the fields of transportation, logistics, healthcare, production, and telecommunication systems. Simheuristics are capable of solving real-world optimisation problems where the simulation component deals with the uncertainty of the model and interacts with the metaheuristic component that, in turn, searches the solution space for a near-optimal solution. These stochastic components can be either located in the objective function or the set of constraints (Chica et al. 2017). Recent research of the simheuristics applications in logistics, transportation, and other supply chain areas is presented by Juan et al. (2018). They highlighted the convenience of combining simulation with metaheuristics for dealing with real-life optimisation problems under uncertainty conditions.

The simheuristic framework approach for the LAS has already been presented by Rabe et al. (2018b). However, in these applications, one of the most relevant challenges is to reduce the search space, considering that the search space will expand when the logistics network increases in size.

2.3 Search Space Reduction based on Pattern Recognition

The authors' LAS is used to recommend the most promising action plans that improve the performance of a given logistics network. Determining these action plans is a combinatorial optimisation problem. In a combinatorial optimisation problem, a solution is constructed from a finite set of objects, and it is considered as an NP-hard problem (Woeginger 2003). In the LAS, the search space consists of all potential actions. The number of possible actions increases exponentially with the logistics network's size. Thus, the number of evaluations needed and, therefore, the response time of the system increases as well. Reducing the size of the search space may help in reducing the number of evaluations in order to reduce the system's run time.

Search space reduction is a method to eliminate undesirable areas in the search space in order to find a solution efficiently. If the initial population, in population-based algorithms, contains some good solutions, the algorithm converges more quickly. However, it is not guaranteed that random solutions have good quality.

One of the approaches in search space reduction to identify and eliminate possible areas in the search space is pattern recognition that is defined by Theodoridis and Koutroumbas (2009) as "the scientific discipline whose goal is the classification of objects into a number of categories or classes". An objective is to discover regularities and similarities hidden in the considered data by automated computer-based algorithms. Pattern recognition has attracted the attention of researchers in the last few decades as a machine learning approach due to its wide spread application areas including medicine, communications, automation, and many others (Singh and Singh 2016). Traditional approaches for pattern recognition are deterministic, statistical, syntactical, template matching, fuzzy logic, and neural-networks-based models (Sharma and Kaur 2013). Ha et al. (2016) present an application using evolutionary algorithms in combination with rule-based pattern recognition for searching stock market prices. Miranda et al. (2014), Reungsinkonkarn and Apirukvorapinit (2014) and Chouhdry et al. (2017) present applications for pattern recognition using EAs to reduce the search space.

3 Approaches for Improving the Performance of Logistics Assistance Systems

An action plan, $\mathbb{A}$, consists of n ordered actions represented as $\mathbb{A} = a_1, a_2, \ldots, a_n$, and an action plan's impact on the logistics network's performance is defined as $R(a_1, a_2, \ldots, a_n)$. An action plan $\mathbb{A}$ causes one or more changes to the simulation model defined as $c(\mathbb{A}) = c(a_1) \cup c(a_2) \cup \ldots \cup c(a_n)$. Those changes may affect one or more entities of the logistics network, e.g., a transport relation or an SKU in a site. Thus, an action a_k is affecting m entities of the logistics network defined as $e(a_k) = e_1, e_2, \ldots, e_m$. Additionally, an action may change one or more attributes of each affected entity, such as the stock level of an SKU at a site or the frequency of a transportation relation. Attributes affected by an action a are expressed as $v(a) = v_1, \ldots, v_m$.

Depending on an action's changes to the logistics network, it is classified as structural or configurational. A structural action causes changes to the structure of the logistics network, such as removing an SKU from a site or establishing a transportation relation between two sites. On the other hand, a configurational action changes an attribute's value of an entity, such as increasing the stock level of an SKU at a site. In specific, a structural action adds or removes one or more entities from the logistics network, whereas a configurational action is changing their attribute values. However, a single action may cause structural *and* configurational changes to the logistics networks, e.g., centralising an assortment in a site.

In this paper, the authors investigate an approach to reduce the number of simulation runs in the authors' LAS. The investigated approach is classified into two parts: firstly, an approach to identify redundant actions in an action plan and second, an approach to identify interchangeable actions. In the following sections, both approaches will be described.

3.1 Identifying Redundant Actions

In this section, the authors are presenting an approach for identifying redundant actions within an action plan. When applying an action plan, redundant actions have no impact on the results of changes to the simulation model. Thus, an action a_i is redundant, if $c(\mathbb{A}_1) = c(\mathbb{A}_2) \mid \mathbb{A}_1 = \mathbb{A}_2 \cup a_i$. In order to identify redundant actions, the following rules have been identified:

For a given $\mathbb{A} = a_1, \ldots, a_i, \ldots, a_j, \ldots, a_k, \ldots a_n$. An action $a_i \in \mathbb{A}$ is redundant if one of the rules applies:

- **Rule 1**: $\exists a_k \in \mathbb{A} : a_i = a_k \mid a_i$ is causing structural changes only and $\forall a_j \in \mathbb{A} : e(a_i) \cap e(a_j) = \emptyset \mid i \leq j < k$.
- **Rule 2**: $\exists a_k \in \mathbb{A} : v(a_i) \subset v(a_k)$, a_i is causing configurational changes only and $\forall v_l \mid v_l \in v(a_i) \cap v(a_k) : v_l$ is set to a fixed value.
- **Rule 3**: $\exists a_k \in \mathbb{A} : e(a_i) \subset e(a_k)$ and a_k is removing entity $e \mid \forall e \in e(a_i) \cap e(a_k)$ from the database.

Actions are redundant if any of these three rules applies. The first rule describes an action plan $\mathbb{A}$ having two or more identical actions, e.g., two actions a_i and a_k each adding a transport relation between site A and site B. If there is no action in the action plan between a_i and a_k that manipulates the same entities as a_i and a_k, then the entities affected by a_i will not be changed in any way until a_k is being executed. Since $a_i = a_k$, a_k will apply the same changes to the entities as a_i did. Thus, a_i is redundant and can be removed from the action plan without changing $c(\mathbb{A})$. This rule can be applied on actions that result only in structural changes to the logistics network. Identical configurational actions can be applied repeatedly to the logistics network and manipulate its state each time a corresponding action is executed. For instance, an SKU X in site A having a stock level of 100 can be manipulated multiple times, e.g., changing its stock by +1 resulting in a stock level of 101, 102, 103 and so on.

Configurational actions are covered by rule 2. Two actions a_i and a_k are in an action plan $\mathbb{A}$ and causes configurational changes to a number of entity's attributes in the logistics network. If the set of parameters that is changed by action a_i is a subset of the parameters changed by a_k, then action a_i is redundant and can be removed from $\mathbb{A}$. This rule may only be applied if all attributes affected by a_k are set to a fixed value, e.g., setting the stock level to 100 or the transport frequency to Monday and Tuesday. An action a_k that is changing the parameter values relatively, e.g., changing the stock level by +10, is not covered. For instance, relative changes by a_i and by a_k may add up resulting in different configurations of the logistics network. Compared to rule 1, this rule is independent of the actions that are in between a_i and a_k.

Rule 3 defines an action a_i as being redundant, if all entities affected by a_i are removed from the logistics network after applying another action a_k with $i < k$. This means, that all changes done by a_i are overwritten by action a_k, e.g., an action a_i changes the stock level of SKU X in site A by +10 and a_k removes SKU X from site A. However, this rule may only be applied if a_k is removing all entities $e(a_i)$.

3.2 Identifying Interchangeable Actions

In each run of the evolutionary algorithm, solutions – represented as an action plan – are generated. Each solution is evaluated, and the algorithm proceeds in searching for better action plans or it terminates. Analysing these action plans leads to consider the actions forming them.

Two action plans $\mathbb{A}_1$ and $\mathbb{A}_2$ may share the same actions arranged in different orders while still having the same impact on the performance of the logistics network. Thus, $R(\mathbb{A}_1) = R(\mathbb{A}_2)$ resulting in $\mathbb{A}_1$ and $\mathbb{A}_2$ as similar action plans. For example, two action plans $\mathbb{A}_1 = a_1, a_2, a_3, a_4, a_5$ and $\mathbb{A}_2 = a_1, a_3, a_2, a_4, a_5$ as shown in Figure 2(a) are considered as two similar action plans. Both action plans have the same actions, with a_2 and a_3 being swapped. Since both action plans have the same effect on the logistics network's performance, they are similar action plans, and the evaluation of one action plan can be used as an evaluation for the other one without running the simulation twice. Actions that can be swapped within an action plan without changing its effect on the logistics network are called interchangeable actions. In Figure 2(b), actions a_2, a_3 and a_4 can be swapped with each other. The evaluation of the first action plan can be used as an evaluation for any of the five action plans that result from swapping the interchangeable actions (Fig. 2(b)).

Identifying interchangeable actions in an action plan results in the possibility of evaluating similar action plans and eliminating the need to run the simulation. Therefore, the authors have defined three rules to identify interchangeable actions.

For a given action plan $\mathbb{A} = a_1, a_2, \dots, a_i, a_{i+1}, \dots, a_n$, interchangeable actions can be defined if at least one of the following rules can be applied.

- **Rule 1:** Actions a_i and $a_{i+1} \mid a_i, a_{i+1} \in \mathbb{A}$ are interchangeable actions $\Leftarrow e(a_i) \cap e(a_{i+1}) = \emptyset \Leftrightarrow a_i$ and a_{i+1} do not affect the same entity.
- **Rule 2:** Actions a_i and $a_{i+1} \mid a_i, a_{i+1} \in \mathbb{A}$ are interchangeable actions if $e(a_i) \cap e(a_{i+1}) \neq \emptyset \Leftrightarrow a_i$ and a_{i+1} affect the same entity e and causing configurational changes only $\Leftarrow$
  a. a_i and a_{i+1} do not affect the same attribute of entity e, or
  b. a_i and a_{i+1} affect the same numeric attribute of entity e (not the entity's ID) changing its value absolutely with the attribute's range limits not being reached after applying either a_i or a_{i+1}.
- **Rule 3:** Actions a_i and $a_{i+x} \mid a_i, a_{i+x} \in \mathbb{A}$ are interchangeable $\Leftarrow$

$\forall a_k \in \mathbb{A} \mid i < k < i + x$: rules 1, 2a or 2b can be applied to a_i, and $a_k \mid a_k \triangleq a_{i+1}$, a_k and $a_{i+x} \mid a_k \triangleq a_i$, and a_i and a_{i+x}.

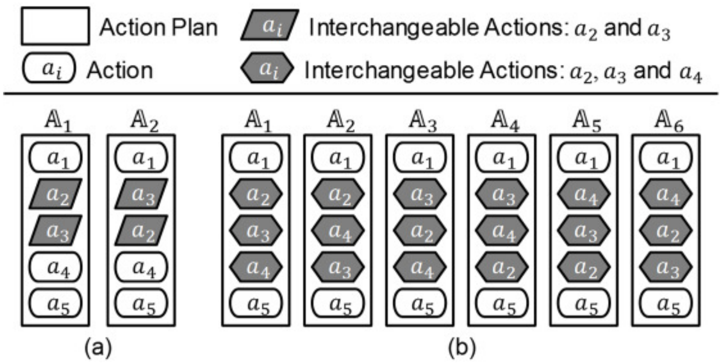

Figure 2: *Similar action plans sharing the same actions and having the same impact on the logistics network: (a) shows two similar action plans having two interchangeable actions, and (b) shows six similar action plans resulting from three interchangeable actions*

Rules 1 and 2 define the conditions for the identifications of subsequent interchangeable actions a_i and a_{i+1} in an action plan $\mathbb{A}$. Rule 1 states, that any two subsequent actions are interchangeable, if they are affecting different entities in the logistics network. This rule can be applied to a subsequent action pair, since the order of any corresponding changes does not affect the resulting state, as long as different entities of the logistics network are adapted. An extension to the first rule is specified in rule 2. If both actions are configurational actions that affect the same entity, they can be interchangeable if one of the following cases applies: Actions a_i and a_{i+1} are changing different attributes of the same entity in the logistics network, or actions a_i and a_{i+1} cause absolute changes to the same numerical attribute value, e.g., increasing the safety stock level by 10. In the second case, the actions are interchangeable if the range limit of any affected attribute is not exceeded after applying any of the actions, and the change is an incremental increase or decrease by a constant value.

Rule 3 states that any two actions a_i and a_{i+x} in an action plan are interchangeable depending on the changes caused by them and the actions between them. The actions a_i and a_{i+x} should be checked with actions between them. If one of the rules 1, 2(a) or 2(b) can be applied to any constructed pairs of a_i and a_{i+x} with actions between them, and actions a_i and a_{i+x}, then actions a_i and a_{i+x} are interchangeable. For example, actions a_2 and a_5 are interchangeable, if the following pairs are interchangeable; a_2 and a_3, a_2 and a_4, a_3 and a_5, a_4 and a_5, and a_2 and a_5.

4 Implementation and Results

Both approaches from the previous chapter are implemented in the EA in the HU as follows: In each iteration, action plans are generated to be evaluated using discrete-event simulation. The defined rules are checked before the evaluation of the action plans. First, all action plans are checked for redundant actions. After removing all redundant actions, the resulting action plan is checked for interchangeable actions. If the action plan contains interchangeable actions, a list of evaluated action plans is checked for similar action plans. The list of evaluated action plans contains all evaluated action plans so far. An action plan, $\mathbb{A}$, and its evaluation $R(\mathbb{A})$ are added to this list after evaluating the action plan by running the simulation. In case a similar action plan has been previously evaluated and stored in the list, its result is retrieved and used as an evaluation for the current action plan. If no interchangeable actions in the action plan or no similar action plan are found, the action plan is applied to the logistics network and evaluated. The result of the action plan is added to the list of evaluated action plans. This process is repeated for all the action plans generated during the iteration. After all action plans are evaluated, the algorithm continues with the next iteration until a termination criterion has been reached.

In order to test the defined approach, experiments were conducted in cooperation with an international trading company on a logistics network consisting of five sites and 30 SKUs. The EA recommends the most promising action plan that decreases the logistics network's cost and increases the service level after reaching 100 iterations. Using a generation size of 20, the total number of evaluations during the algorithm run becomes 2,000. Since this paper aims to present the approach to reduce the number of simulation runs, the relative reduction in the number of simulation runs is plotted in Figure 3. In the experiments, crossover and mutation rates were varied. In Figure 3, the mutation rate has the greatest effect on the reduction of simulation runs. The probability of generating similar action plans is reduced with higher mutation rates that introduce new actions to the action plans.

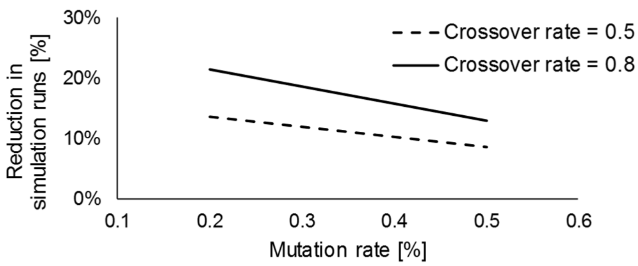

Figure 3: *Percental reduction in simulation runs in different crossover and mutation rates*

The number of simulation runs is reduced by 12.8 % by defining similar actions when the crossover rate is 0.8, and the mutation rate is 0.5. On the other hand, removing redundant actions results in a new action plan and a reduction of 12.3 % at the same crossover and mutation rates. Removing redundant actions from an action plan shortens the action plan, which is a new action plan to the algorithm. On the other side, the shortened action plan provides a place to introduce new actions to the action plan. Thus, new action plans can be generated that may constitute better solutions and help in exploring the search space.

5 Conclusion and Outlook

The proposed approaches aim to increase the efficiency and effectiveness of a LAS. The approaches define redundant and interchangeable actions. Redundant actions can be removed from an action plan and be used to form new action plans. Interchangeable actions are used to identify similar action plans. The evaluation of an action plan can be used to assess any similar action plan.

For further work, the authors will extend the usage of redundant actions to form new action plans and study new rules to be added to the identified approaches. Thus, approaches can be defined to replace the redundant actions in an action plan. These approaches can enable greater utilisation of the iterations in the algorithm run in the considered LAS. For further research, a machine learning approach can be utilised to identify further interchangeable actions

References

Blutner, D.; Cramer, S.; Krause, S.; Mönks, T.; Nagel, L.; Reinholz, A.; Witthaut, M.: Assistenzsysteme für die Entscheidungsunterstützung. In: Buchholz, P.; Clausen, U. (Eds.): Große Netze der Logistik. Berlin, Heidelberg: Springer 2009, pp. 241–270.

Chica, M.; Juan, A.A.; Cordon, O.; Kelton, D.: Why Simheuristics? Benefits, Limitations, and Best Practices When Combining Metaheuristics with Simulation. SSRN Electronic Journal (2017), 10.2139/ssrn.2919208.

Chouhdry, Z.R.; Hasan, K.M.; Raja, M.A.: Design of Reduced Search Space Strategy Based on Integration of Nelder–Mead Method and Pattern Search Algorithm with Application to Economic Load Dispatch Problem. Neural Computing and Applications 71 (2017) 1, pp. 1–13.

Ha, M.H.; Lee, S.; Moon, B.R.: A Genetic Algorithm for Rule-based Chart Pattern Search in Stock Market Prices. In: Neumann, F.; Friedrich, T.; Sutton, A. (Eds.): Proceedings of the Genetic and Evolutionary Computation Conference - GECCO '16. New York (USA): ACM Press, 2016, pp. 909–916.

Juan, A.A.; Rabe, M.: Combining Simulation with Heuristics to Solve Stochastic Routing and Scheduling Problems. In: Dangelmaier, W.; Laroque, C.; Klaas, A. (Eds.): Simulation in Produktion und Logistik - Entscheidungsunterstützung von der Planung bis zur Steuerung. Paderborn: HNI-Verlagsschriftenreihe, 2013, pp. 641–649.

Juan, A.A.; Kelton, W.D.; Currie, C.S.M.; Faulin, J.: Simheuristics Applications: Dealing with Uncertainty in Logistics, Transportation, and other Supply Chain Areas. In: Rabe, M.; Juan, A.A.; Mustafee, N.; Skoogh, A.; Jain, S.; Johansson, B. (Eds.): Proceedings of the 2018 Winter Simulation Conference. Piscataway (USA): IEEE 2018, pp. 3048–3059.

Ławrynowicz, A.: A Survey of Evolutionary Algorithms for Production and Logistics Optimization. Research in Logistics and Production 1 (2011), pp. 57–91.

Liebler, K.; Beissert, U.; Motta, M.; Wagenitz, A.: Introduction OTD-Net and LAS: Order-to-Delivery Network Simulation and Decision Support Systems in Complex Production and Logistics Networks. In: Pasupathy, R.; Kim, S.H.; Tolk, A.; Hill, R.; Kuhl, M.E. (Eds.): Proceedings of the 2013 Winter Simulation Conference. Piscataway (USA): IEEE, 2013, pp. 439–451.

Miranda, R.D.; Montevechi, J.A.; Silva, A.F.; Marins, F.A.: A New Approach to Reducing Search Space and Increasing Efficiency in Simulation Optimization Problems via the Fuzzy-DEA-BCC. Mathematical Problems in Engineering 2014 (2014), pp. 1–15.

Rabe, M.; Dross, F.; Schmitt, D.; Ammouriova, M.; Ipsen, C.: Decision Support for Logistics Networks in Materials Trading Using a Simheuristic Framework and User-generated Action Types. In: Wenzel, S.; Peter, T. (Eds.): Simulation in Produktion und Logistik 2017. Kassel: Kassel University Press, 2017, pp. 109-118.

Rabe, M.; Ammouriova, M.; Schmitt, D. (2018a) Utilizing Domain-specific Information for the Optimization of Logistics Networks. In: Rabe, M.; Juan, A.A.; Mustafee, N.; Skoogh, A.; Jain, v; Johansson, B. (Eds.): Proceedings of the 2018 Winter Simulation Conference. Piscataway (USA): IEEE, 2873-2884.

Rabe, M.; Schmitt, D.; Ammouriova, M. (2018b) Improving the Performance of a Logistics Assistance System for Materials Trading Networks by Grouping Similar Actions. In: Rabe, M.; Juan, A.A.; Mustafee, N.; Skoogh, A.; Jain, S.; Johansson, B. (Eds.): Proceedings of the 2018 Winter Simulation Conference. Piscataway (USA): IEEE, 2861-2872.

Reungsinkonkarn, A.; Apirukvorapinit, P.: Search Space Reduction of Particle Swarm Optimization for Hierarchical Similarity Measurement Model. In: 11th International Joint Conference on Computer Science and Software Engineering. Piscataway, N.J.: IEEE, Chon Buri, Thailand, 14-16 May, 2014, pp. 23–27.

Sharma, P.; Kaur, M.: Classification in Pattern Recognition. A Review. International Journal of Advanced Research in Computer Science and Software Engineering 3 (2013) 4, pp. 298–306.

Singh, A.; Singh, P.: Role of Hybrid Evolutionary Algorithms in Pattern Recognition. International Journal of Computing and Technology 3 (2016) 4, pp. 244–249.

Tako, A.; Robinson, S.: The Application of Discrete Event Simulation and System Dynamics in the Logistics and Supply Chain Context. Decision Support Systems 52 (2012) 4, pp. 802–815.

Theodoridis, S.; Koutroumbas, K.: Pattern Recognition. San Diego, CA. USA: Elsevier Inc. 2009.

Turban, E.: Decision Support and Business Intelligence Systems, 9. ed. Upper Saddle River, N.J.: Pearson Prentice Hall 2007.

Woeginger, G.J.: Exact Algorithms for NP-hard Problems: A Survey. In: Jünger M.; Reinelt G.; Rinaldi G (Eds.): Combinatorial Optimization — Eureka, You Shrink! Papers Dedicated to Jack Edmonds. Berlin, Heidelberg: Springer 2003, pp. 185–207.