Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A.A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

UTILIZING DOMAIN-SPECIFIC INFORMATION FOR THE OPTIMIZATION OF LOGISTICS NETWORKS

Markus Rabe
Majsa Ammouriova

Department IT in Production and Logistics
TU Dortmund University
Leonhard-Euler-Str. 5
Dortmund, 44227, GERMANY

Dominik Schmitt

Graduate School of Logistics
TU Dortmund
Leonhard-Euler-Str. 5
44227 Dortmund, GERMANY

ABSTRACT

Continuously maintaining a logistics network (LNW) in good condition is a challenging task for decision makers. For purposes of improving an LNW's performance, promising actions need to be identified, such as the centralization of a stock keeping unit (SKU). In order to support the decision maker, the authors have developed a logistics assistance system (LAS) based on discrete-event simulation. With an increasing size of the LNW, the response time of such an LAS increases exponentially. In this paper, the authors present an approach for utilizing domain-specific information to guide the search for promising actions and, therefore, reduce the LAS's response time. The given examples show that the LAS's response time can be decreased. For example, the approach reduces the number of iterations needed by an evolutionary algorithm to converge.

1 INTRODUCTION

Decision makers aim to continuously increase the performance of logistics networks, which may be defined by key performance indicators (KPIs). Examples of such KPIs are the total cost of the logistics network or its service level (Rushton et al. 2006; Brandimarte and Zotteri 2007). The total cost involves costs of various logistics activities, e.g., handling costs, stock costs, or transportation costs. The service level represents the overall degree of customer satisfaction, such as on-time delivery, product characteristics, and price (Ghiani et al. 2013). In order to affect KPIs and, therefore, the performance of the logistics network, actions need to be applied. Such actions in logistics networks may be decreasing the stock of a stock keeping unit (SKU) at a site, increasing a transport relation's frequency or centralizing an assortment at a central warehouse, where an assortment is a collection of SKUs with similar application area or similar characteristics.

An action may affect various areas of the logistics network, such as the transport, the handling of SKUs at a site, or their distribution to the customers (Ghiani et al. 2013). In addition, actions may have contrary effects on several KPIs, e.g., increasing the stock of an SKU at a site may improve the service level while increasing the stock costs at the same time (Rushton et al. 2006). Thus, finding the most promising actions for increasing the overall performance of the logistics network is a challenging task.

In order to address this problem, the authors have developed a logistics assistance system (LAS) for determining and suggesting promising action sets, a number of actions, to the decision makers (Rabe et al. 2017c). The LAS is based on a simheuristic framework, which combines a heuristic algorithm and discrete-event simulation (DES) (Juan and Rabe 2013). The heuristic algorithm is trying to find promising actions and DES is used to evaluate the impact of these actions on the logistics network. The search and evaluation of actions can be performed iteratively. With an increased size of the logistics network and a raising number of possible actions, the overall response time of the logistics assistance system increases as well. This may lead to an unfeasible delay between querying the LAS and getting a response.

Therefore, the authors are examining an approach to guide the LAS's search for the most promising action set by utilizing domain-specific information (DSI). The DSI is added to an action including information about its success, frequency, type, and correlation with other actions. The authors investigated the utilization of actions' success resulting in better results (Rabe et al. 2018). In this research, the authors include additional DSI to improve the LAS's results further.

This paper is organized as follows: Section 2 gives an overview of the related work. Section 3 introduces domain-specific information and its integration in actions. Section 4 describes the utilization of DSI in the heuristic algorithms. In section 5, experiments and their results are described. Section 6 presents an outlook.

2 RELATED WORK

2.1 Domain-specific Modeling Languages

In general, a modeling language can formally describe either information, knowledge, or systems in a structured way. The benefits of using a modeling language are a less complex modeling process and an increased understanding of the modeled system. The language's structure is defined by a consistent set of rules. These rules are used for interpretation of the modeling language's constructs. In addition to well-formed rules, a modelling language consists of its syntax (notation) and its semantics (meaning) (Harel and Rumpe 2004). A modeling language, expressed in a textual or graphical way, can be used to describe a system or its behavior in an abstract way. A prominent example of such a modeling language is UML (Object Management Group 2017).

Modeling languages can be divided into two categories. The first category is formed by general-purpose (modeling) languages, which can be applied for a wide range of different problems. The second category of modeling languages is restricted to well-defined problem domains. These languages are called domain-specific (modeling) languages (DSL), little languages (van Deursen and Klint 1998) or microscopic languages (Bentley 1986). Van Deursen et al. (2000) define a DSL as a "programming language or executable specification language that offers, through appropriate notations and abstractions, expressive power focused on, and usually restricted to, a particular problem domain". By restricting the language to a specific problem domain, only a small set of abstractions and notations is needed. Thus, using a domain-specific language, instead of a general-purpose language increases its usability, productivity, reliability and maintainability (Kieburtz et al. 1996). Examples of DSLs are SQL, BNF or HTML.

In conclusion, by using a DSL a better abstraction of the modeling process can be achieved.

2.2 Decision Support for Logistics Networks

Logistics assistance systems (LAS) or decision support systems in logistics are used for supporting decision makers in critical decisions (Blutner et al. 2007; Kuhn et al. 2008). In general, the terms LAS and decision support system are used synonymic when talking about the domain of logistics. The term DSS is the most widely used one in this specific domain (Kengpol 2008). However, the authors are using the term logistics assistance system or LAS throughout the paper to describe this type of systems in order to highlight the system's domain.

In the international literature, there are several simulation-based approaches of logistics assistance systems. A simulation-based LAS for the disposition in global supply chains is described by Deiseroth et al. (2008). Bockholt et al. (2011) describe an approach for the collaboration in global supply chains. Both approaches are mainly focusing on global supply chains in the automotive sector. An overview of an order-to-delivery network simulation and logistics assistance system for production and logistics networks in the automotive sector is given in Liebler et al. (2013).

2.3 Logistics Assistance System for Logistics Networks in Materials Trading

The authors have developed a logistics assistance system for logistics networks in materials trading. The LAS's architecture is shown in Figure 1. In this section, a brief overview of the working principles is provided. For an extended overview of the logistics assistance system, the reader is kindly referred to previous publications of the authors (Rabe et al. 2017c; Rabe et al. 2018).

Companies typically use some software for storing and utilizing data of their logistics network, such as transactional systems, data warehouses, or KPI monitoring systems. The company's data are extracted, transformed and loaded from the transactional system into the data warehouse (DWH) periodically. The data in the DWH are used for calculating KPIs that indicate the logistics network's performance. KPIs are monitored by KPI monitoring systems (KPIMS). On a regular basis, the KPIMS are analyzing and comparing the KPIs with predefined criteria. The results from this comparison are aggregated into a KPI report which can be accessed by decision makers. In addition, certain conditions may trigger a KPI alert, e.g., if a KPI leaves a predefined range. The KPI alert contains two kinds of information, the reason triggering the alert and a list of possible actions. Actions, e.g., increasing the transport frequency of a transport relation or shifting SKUs from one site to another site, are changing the logistics network's state. These action proposals are focused on the corresponding KPI in order to bring back the KPI towards its predefined target. However, an action suggestion may have interdependencies with other proposed actions, which may lead to a negative effect on the overall performance of the logistics network. Therefore, action suggestions will be transferred to the LAS for further investigation, as seen in Figure 1.

The LAS uses a simheuristic approach (Juan and Rabe 2013) for determining and evaluating promising actions for improving the logistics network's performance. A data-driven DES model is used as an evaluation function. To have an up-to-date model of the real-world logistics network, the company's data are automatically extracted from the transactional system and the DWH. Subsequently, the model builder is transforming the extracted data and loading them into a data model, which is stored in a MySQL database. If necessary, the data model can be updated at any time in order to provide the LAS with the most recent state of the real logistics network. The data from the database are used to create the simulation model. Experiments can be run on the simulation model to evaluate the logistics network's performance. The simulation results are written to the database. These results are transferred into a shadowed data warehouse (SDWH) from where they can be used for KPI calculations on the simulated scenario. By shadowing the data warehouse and the KPI logic, a strict separation between real-world data and simulation data can be ensured. The KPIs are provided to the heuristic unit (HU) for further investigation.

Based on the suggested KPIs, the HU is searching for promising actions to improve the overall performance of the logistics network. For purposes of determining promising actions, the heuristic unit needs access to all actions that can be potentially applied in the logistics network. Thus, the available actions are creating the search space for the HU. To access the search space, the HU is linked to the action type directory, which stores all available action types. An action type is a generalization of similar actions, e.g., an action type may describe an increase of the transport frequency between two arbitrary locations in a generic way. Corresponding actions are specified by defining concrete locations, e.g., increasing the transport frequency from site A to site B or from supplier S to site C, whereby A, B, C and S are specific locations of the logistics network. Hence, actions can be derived from action types by adding distinct parameter values to the action type. Based on the current state of the logistics network and the available action types, the LAS calculates corresponding actions. Being connected to the action type directory, the heuristic unit has access to all action types and, therefore, to all actions.

The heuristic unit is examining the search space for promising actions regarding the overall performance of the logistics network. A more detailed description of the HU's implementation is provided in chapter 4 as well as in Rabe et al. (2017b). The actions selected by the HU are forwarded to the execution engine. The execution engine decomposes the actions into changes to the data model. These changes will be applied to the data of the underlying database in order to manipulate the state of the logistics network.

The action's effect will be evaluated by instantiating the simulation model with the modified data and running a simulation experiment. The process of determining and evaluating promising actions can be run iteratively. When reaching certain termination criteria, such as a certain number of iterations or a specific quality for the best found solution, this process stops. The most promising actions are addressed as suggested actions and provided to the user. These suggested actions can be used as a decision support for adjustments to the real logistics network.

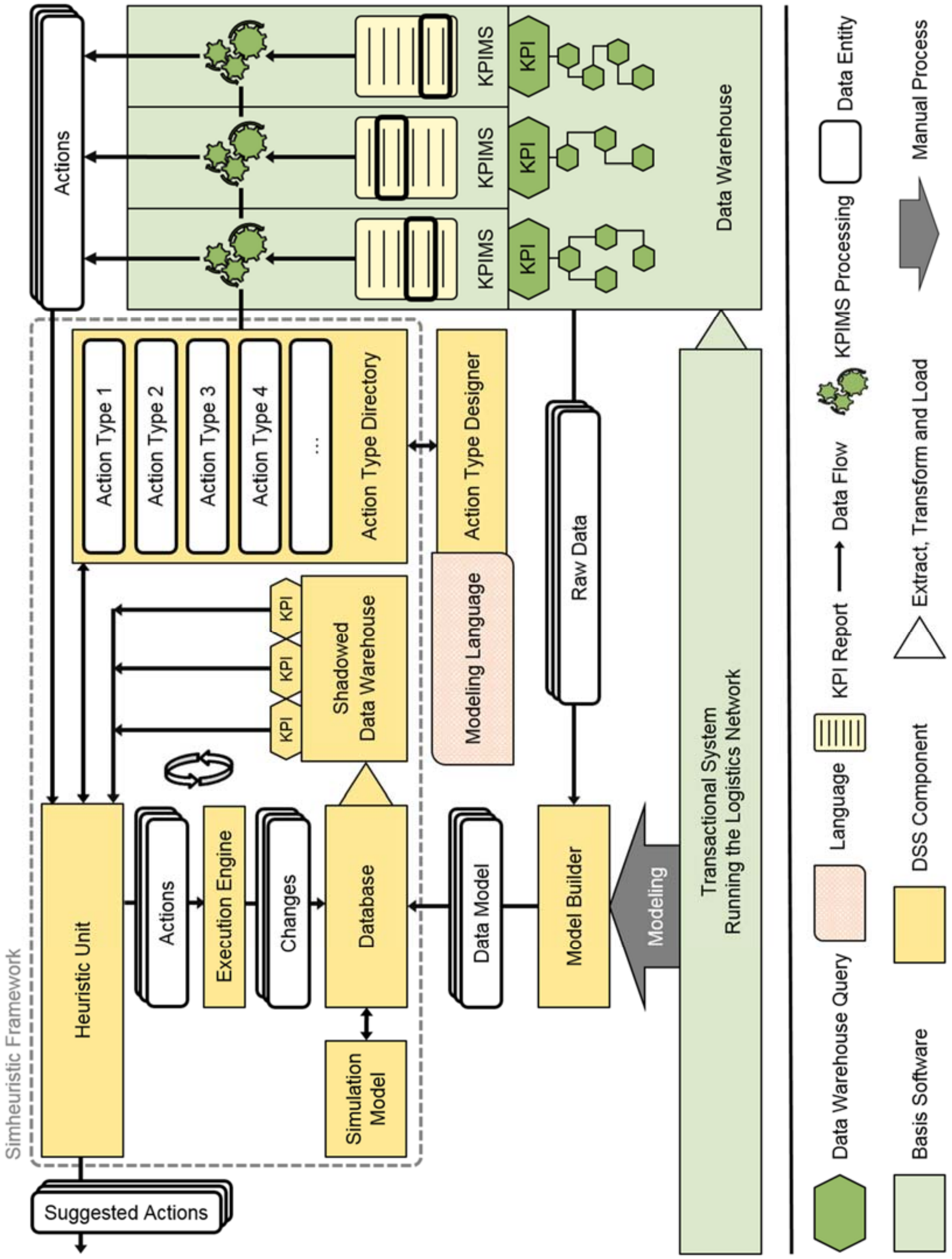

Figure 1: Architecture of the logistics assistance system, based on Rabe et al. (2017c).

In order to increase the logistics assistance system's flexibility, users may add new action types to the system. Therefore, users have access to the action type designer. The action type designer is an interface, where the user can access predefined constructs in order to specify new action types. These constructs, described in a domain-specific modeling language, are hiding the technical implementation of changes to the database and abstracting them to a level that is more convenient for the application. Hence, defining action types with predefined constructs may ease the modeling process. A more detailed description of this process is presented in Rabe et al. (2017c).

By adding new action types to the LAS or by increasing the size of the logistics network, the number of possible actions and, therefore, the size of the search space are increased. Examining the entire search space is very time consuming and thus, unfeasible in most situations. In order to reduce the search time without deteriorating the quality of the provided actions, the authors are experimenting with an approach to integrate additional information into the modeling process of action types, as described in chapter 3.

2.4 Algorithms Implemented in the Heuristic Unit

The heuristic unit is searching the search space for the most promising action set, based on the current state of the logistics network. In the current realization of the HU, an Evolutionary Algorithm (EA) and Deep Reinforcement Learning (DRL) are implemented.

The *Evolutionary Algorithm* is a stochastic, iterative, population-based search and optimization algorithm that mimics the mechanics of the natural evolutionary process (Bozorg-Haddad et al. 2017). Population-based search means that the algorithm explores a population while searching for a candidate solution. This type of problem is defined as a combinatorial optimization problem, where a solution is constructed from a finite number of objects (Pétrowski and Ben-Hamida 2017). It provides good but non-optimal solutions to problems that cannot be solved by exact methods (Pétrowski and Ben-Hamida 2017).

The EA starts with the initialization of an initial generation, a number of individuals. Individuals, also called chromosomes, represent solutions of the considered problem (Pétrowski and Ben-Hamida 2017). An individual is divided into genes, representing the decision variables' values. A position of a gene in the individual is called locus. The generated individuals are evaluated by assigning a fitness value. This value assesses the quality of the individual. Hence, it is used to differentiate good and bad solutions (Bozorg-Haddad et al. 2017). Subsequent generations are formed by selecting individuals from previous generations based on their fitness value and applying variation operators (Pétrowski and Ben-Hamida 2017). These variation operators are crossover, also called recombination, and mutation. In crossover, genes of selected-mating individuals are exchanged. Thus, new individuals are formed and the current solutions are exploited with a view to find better ones (Ahn 2006). On the other side, mutation perturbs individuals slightly by changing an arbitrary gene in an individual. It helps in getting away from the local optima and thus has greater exploratory power. The newly generated individuals are evaluated. Afterwards, the evolution and evaluation of new generations continue until a termination criterion is met. Thus, EA is an algorithm for exploitation and exploration (Spears 2000).

In *Reinforcement Learning*, an agent learns from consequences of applied actions to an environment. RL can be applied when the problem could be formulated as a finite Markov decision process (Mnih et al. 2013). The agent considers the environment's state and can take actions affecting this state based on defined goals (Sutton and Barto 1998). The environment represents the studied system and the agent explores a search space of actions by using trial and error (Stockheim et al. 2003). When applying an action, the agent gets a scalar reward signal that indicates the quality of the action regarding the defined goal of the RL. In addition, a policy is used to define the agent's way of behaving (Sutton and Barto 1998). This policy maps the perceived state of an environment to actions. In an epsilon-greedy policy, random actions are selected with an *epsilon* probability. The action with the greatest expected reward is selected with a probability of 1-*epsilon*. Over a predefined number of steps, epsilon's value is decreasing from 1.0 to a predefined lower border, for instance 0.1. The agent's experience is formed from its interaction with the environment and neural networks can provide the ability to generalize from its experience (Sutton and Barto 1998). Combining RL with a deep neural network, such as a convolutional neural network, forms a Deep

Reinforcement Learning (DRL). In the DRL, the convolutional neural network can be trained with a Q-learning algorithm. The Q-learning algorithm can learn a policy through its own experience and can be used to solve a Markov decision process (Mnih et al. 2015). Mnih et al. (2013) called training deep neural network with Q-learning as Deep Q-network (DQN), and used it to train an agent to play Atari games.

In DRL utilizing DQN, an agent explores the search space of actions based on the current environment state and its gained knowledge. Each environment state and action pair have an associated expected return $Q_{\pi}(s,a)$, resulting from applying action a in state s while following policy π. Then, Q-learning is applied as in equation (1) where α is a learning rate parameter, r is the reward, and γ is the discount rate parameter at time t (Mnih et al. 2015).

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_{a} Q(s_{t+1}, a_t) - Q(s_t, a_t)] \tag{1}$$

The implementation of the DQN is described in Rabe et al. (2017a).

3 INTEGRATION OF DOMAIN-SPECIFIC INFORMATION FOR MODELING ACTIONS

A key principle of the proposed logistics assistance system is the usage of a data-driven simulation, in which the simulation model gets instantiated from the data model. The data model is stored in a MySQL database that consists of two parts. The first component is the data structure that describes the structure of the database, e.g., tables and relations between those tables. The second part is the data itself that describe the current state of the logistics network. Action types and, therefore, corresponding actions are implemented as changes to the data of the simulation's database instead of affecting the simulation model directly. The authors presented a method for creating and applying action types for DES of logistics networks in Rabe et al. (2017c).

Depending on the data model's structure, an action type may affect different parts of the data structure in various ways. E.g., when centralizing an assortment at a site, corresponding SKUs may be added to the central warehouse, whereas some of the SKUs may be removed from other affected sites. Additionally, the sourcing of corresponding SKUs in affected sites may need to be adjusted in order to keep the data model in a valid state. A complex action type can be modeled by composing existing action types, e.g., an action type representing the centralization of an assortment at a site may consist of several sub-action types. Such sub-action types may describe the required changes for adding or removing SKUs from a site, adjusting the sourcing of SKUs, adding new or removing obsolete transport relations. The relation between an action type and the affected parts of the data structure, as well as the kind of changes and the integrated sub-action types are stored as functional data within the action type.

Action types may be modeled and used by different users. Therefore, additional information for ensuring the user's understanding of an action type is required. Such informational data can be added to the action type, including a name or a description or the required input parameter of an action type.

Besides the functional and informational data, the authors propose to integrate domain-specific information into the definition of action types. When a user models an action type, DSI based on experience from the real-world logistics network can be added. Likewise, the HU may calculate computational domain-specific information based on experience gathered throughout previous simulation experiments and integrate these information into the action types as well.

The authors propose to use different kinds of DSI, as pictured in Figure 2. These information can be utilized in the search for promising actions (see chapter 4).

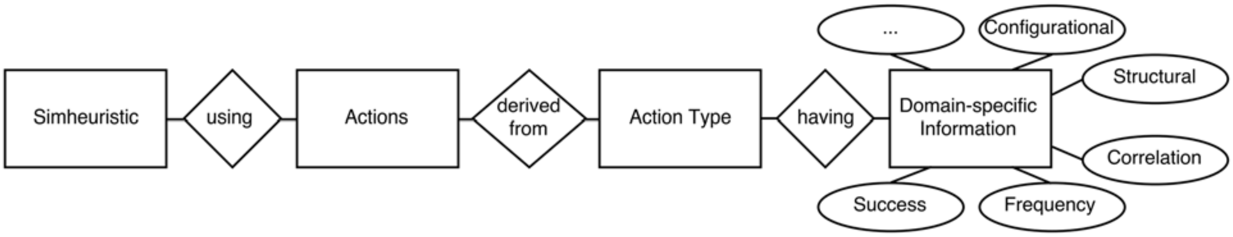

Figure 2: Relations between simheuristic, actions, action types and domain-specific information.

First, action types can be segmented into two types: either structural or configurational. A structural action type represents changes affecting the structure of the logistics network, such as shifting an assortment to a site, adding transport relations, reallocating customers, or removing sites. Changes to the state of the logistics network, e.g., adjusting the safety stock of an SKU at a site, increasing the frequency of a transport relation, or reducing the minimum order quantity of an SKU are represented by configurational action types. When applying a configurational action, e.g., for increasing the transport frequency of a transport relation first, and then applying a structural action, e.g., for removing this transport relation may undo the prior adjustment of the frequency. By differentiating between structural and configurational actions it is possible to prevent a structural action to undo the changes of a previous configurational action, for instance by applying all structural actions before adjusting the state of the logistics network.

Furthermore, additional domain-specific information such as the success or use frequency may be added to an action type's definition. The use frequency of an action type indicates how common corresponding actions have been applied in the past. An action type which has been applied 100 times may be a more promising candidate for further investigation than another action type that has only been used five times. The success of an action type indicates the average effect on the network's profitability when applying corresponding actions, e.g. actions A1, A2, and A3 for reducing the stock of an assortment in a warehouse may reduce the logistics network's total cost by 0.01%, 0.035%, and 0.045% respectively. This leads to an average success value of 0.03% for the action type indicating its expected impact on the total costs of the logistics network. The success and frequency of an action type are considered for a given time period, e.g. the last business year.

Another DSI may be the correlation between action types. When applying an action, the correlation indicates the influence on correlated actions for further investigations. For example, when centralizing an assortment in a site, increasing the safety stock level of corresponding SKUs in that site may be a promising candidate. This results in a positive correlation between the corresponding action types. Whereas, decreasing the stock of affected SKUs in that site may be a contrary action which leads to a negative correlation. The correlation is a directed relation between action types and consists of two parameters. The first parameter is a list of IDs referencing the correlating action types and the second parameter is a correlation factor. Action types can have multiple correlations, whereas some of them may be more promising than others. The correlation factor is used in order to address this issue by representing the correlation's intensity.

The DSI can be utilized by the heuristic unit to guide and improve the search for promising actions, as described in the following chapter.

4 USAGE OF DOMAIN-SPECIFIC INFORMATION IN THE HEURISTIC UNIT

In order to utilize the DSI, the algorithms in the heuristic unit need to be adapted. In the following sections, the adaptation of the EA and DRL is presented.

4.1 Domain-specific Information Usage in the Heuristic Unit

The success value is used to assign a score value to each action type. The score values are scaled to (0,1] according to equation (2), where $v(j)$ is the success value and $score(j)$ is the scaled success value of an action type j. The min and max variables in the equation represent the minimum and the maximum values of the success, respectively.

$$score(AT) = (v(AT) - \min + 1)/(max - \min + 1) \tag{2}$$

The probability of selecting action a derived from the corresponding action type j is calculated based on equation (3), with n_{AT} and na_i being the number of considered action types and the number of actions derived from action type i, respectively. In order to avoid a probability of 0 for selecting a specific action, the authors propose to define the score values' range to be bigger than zero.

$$Prob(a/AT = j) = score(j)/\sum_i^{n_{AT}}(score(i) \cdot na_i) \tag{3}$$

In addition to the success value, the type of an action type is considered in the HU's implementation. This information has been added to the search algorithm to alter the actions' selection. The probability of selecting a structural action is first set to be 0.9. Thus, the probability of selecting a configurational action is set to 0.1. The probability of selecting structural actions will decrease with each action being added to the action set according to equation (4), with x denoting the position and y defining the number of total actions in the respective action set.

$$Prob(structural\_action) = 0.9 - 0.8 \cdot ((x - 1)/(y - 1)) \tag{4}$$

4.2 Adapted Deep Reinforcement Learning Algorithm

Typically, an agent in DRL is selecting actions randomly or based on the agent's experience. In the adapted algorithm, the selection of actions is additionally influenced by their score value and type. Firstly, the algorithm determines to select a structural or configurational action according to the probability calculated in equation (4). Then, a specific action is selected based on the action's score value as in equation (3).

4.3 Adapted Evolutionary Algorithm

In the EA, an individual represents an action set of selected actions ordered by their order of implementation. To utilize DSI, EA's operations, such as the initialization, the crossover, and the mutation operations have been adapted.

Employing DSI modifies the random selection of actions for the initialization. The probability of selecting a structural or configurational action is given in equation (4) with x representing the locus of the individual and y defining the individual's size. Then, the selection of a specific action depends on its calculated score as in equation (3).

The second altered operation in the EA is the crossover of mating individuals. To select the pair of actions from the mating individuals, a locus within the individual's size is randomly selected. The actions in this locus undergo the crossover operation, depending on their DSI. If both actions have the same type, structural or configurational, the score value of the two actions influences the domination of the higher-score-value action or the exchange of the two actions between the individuals. Considering two actions a_1 and a_2, action a_1 having the greater score value dominates by replacing action a_2 in its individual with a probability calculated in equation (5). Otherwise, both actions are exchanged between the individuals.

$$Prob(a_1) = 1 - score(a_2)/score(a_1) \tag{5}$$

If one action is a structural action and the other action is a configurational one, the crossover operation is influenced by the selected locus. For each locus, a structural action's probability is calculated based on equation (4) whereas, the configurational action's probability is complemented. If the structural action's probability is greater than the configurational action's probability, the structural action dominates with its probability. Otherwise, both actions are swapped between the individuals. If the configurational action's probability is greater, the configurational action dominates with its probability. Actions are exchanged otherwise.

In the mutation operation, an action is randomly selected to be replaced by another action from the search space. Actions with a lower score value have a greater probability of being replaced. Based on the action's locus, a structural or a configurational action is selected from the search space as described in the initialization operation.

5 COMPARISION OF APPROACHES

5.1 Designed Experiments

In order to evaluate the effect of utilizing DSI, the authors compare between the original and the adapted algorithms (see sections 4.2 and 4.3), regarding the LAS's performance. For the purpose of determining the performance of the LAS, the convergence speed of the used algorithm and the quality of the best found solution are taken into account.

The search space is constructed from actions derived from the action types listed in Table 1. The EA's parameters were set as follows: generation size 40, individual size 10, crossover probability 0.8, and mutation probability 0.2. The DRL's parameters were set as found in Mnih et al. (2015), and an episode represents an execution of 6 actions. The experiments were run on a small-sized logistics distribution network consisting of 5 sites and 30 SKUs categorized into 6 assortments.

Table 1: Action types used to derive actions in the experiments.

Structural action types	Configurational action types
Centralize SKU Centralize assortment Decentralize SKU Decentralize assortment	Increase stock level Decrease stock level Change disposition attribute of an SKU in a site

5.2 Results and Discussion

The EA's experiment results are shown in Figure 3. The figure shows that the adapted EA, utilizing DSI, found better solutions regarding costs and service level compared to the original algorithm. In addition, utilizing DSI enables the algorithm to converge faster. Thus, the quality of the best found solutions and the convergence speed of the EA have been improved by utilizing DSI.

The average reward in the DRL experiment is plotted in Figure 4. The figure shows the average reward per episode for the original and the adapted DRL. It is seen that the adapted DRL can have a promising advantage over the original DRL, and more research and investigation is motivated. Additionally, the original DRL tends to select three structural actions at the beginning of the episode, and afterwards only configurational actions are selected.

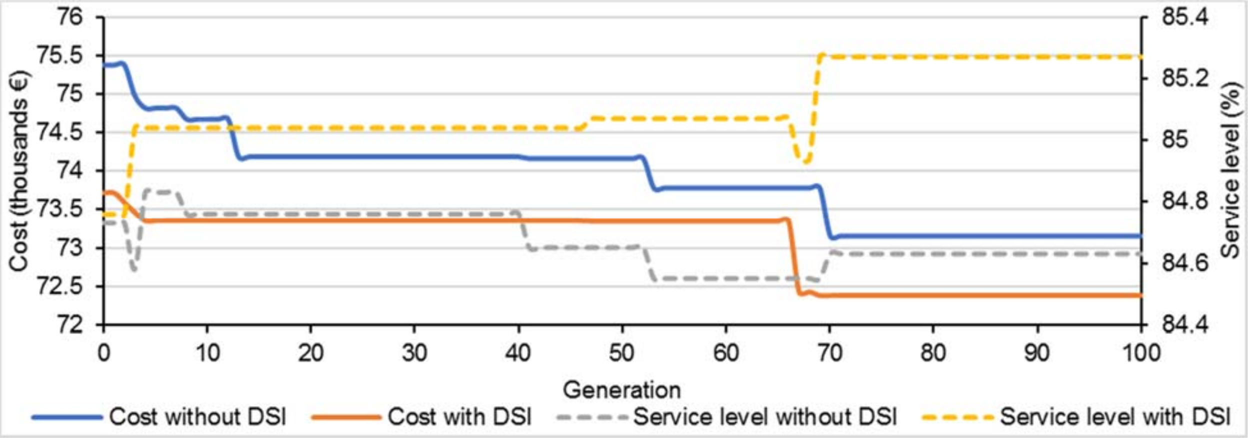

Figure 3: Best found cost and service level in each generation for original and adapted EA.

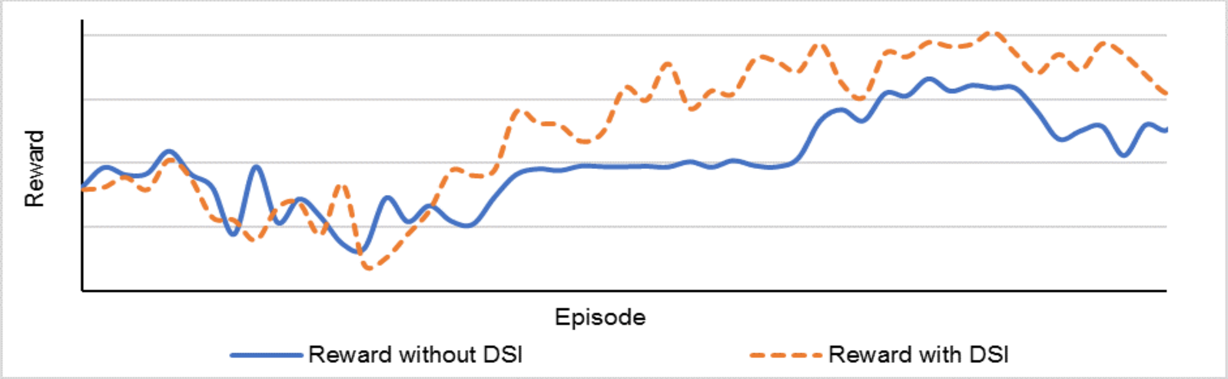

Figure 4: Average reward in original and adapted DRL experiments.

6 CONCLUSION AND OUTLOOK

Using the presented approach, the authors aim to improve the quality and response time of an existing logistics assistance system for actions in logistics networks. Therefore, the most relevant domain-specific information, such as success, frequency, type, and correlation, have been identified, defined, and integrated into the definition of actions. Additionally, the search algorithms of the system, a Deep Reinforcement Learning and an Evolutionary Algorithm, have been adapted in order to utilize the domain-specific information success and type in the search for promising actions. Experiments showed that the utilization of domain-specific information in an EA decreases the number of generations required to converge and improves the recommended solution. The implementation of DSI in DRL is promising and motivates further research in this direction.

For further work, the authors will extend their presented approach by including DSI in the definition of actions, such as the action's effect on the service level or on corresponding KPIs of the logistics network. Furthermore, the authors will investigate the possibility of calculating and integrating computational DSI in addition to the information based on the real logistics network's data. For further improvements in the search of promising actions, the utilization of additional domain-specific information, such as the frequency and the correlation between actions, seems promising.

ACKNOWLEDGMENTS

Special thanks to the Graduate School of Logistics, TU Dortmund, thyssenkrupp, and German Jordanian University for supporting this research.

REFERENCES

Ahn, C. W. 2006. *Advances in Evolutionary Algorithms: Theory, Design and Practice*. Berlin, Heidelberg: Springer.

Bentley, J. 1986. "Programming Pearls: Little Languages". *Communication of the ACM* 29(8):711–721. DOI: 10.1145/6424.315691.

Blutner, D., S. Cramer, S. Krause, T. Mönks, L. Nagel, A. Reinholz, and M. Witthaut. 2007. „Assistenzsysteme für die Entscheidungsunterstützung". Technical Report No. 06009. Dortmund: TU Dortmund. ISSN 1612-1376.

Bockholt, F., W. Raabe, and M. Toth. 2011. "Logistic Assistance Systems for Collaborative Supply Chain Planning". *International Journal of Simulation and Process Modelling* 6(4):297–307.

Bozorg-Haddad, O., M. Solgi, and H. A. Loaiciga. 2017. *Meta-heuristic and Evolutionary Algorithms for Engineering Optimization*. Hoboken, New Jersey: John Wiley & Sons.

Brandimarte, P., and G. Zotteri. 2007. *Introduction to Distribution Logistics*. Hoboken, New Jersey: Wiley.

Deiseroth, J., D. Weibels, M. Toth, and A. Wagenitz. 2008. „Simulationsbasiertes Assistenzsystem für die Disposition von globalen Lieferketten". In *Advances in Simulation for Production and Logistics Applications*, edited by M. Rabe, 41–50. Stuttgart, Germany: Fraunhofer IRB.

Ghiani, G., G. Laporte, and R. Musmanno. 2013. *Introduction to Logistics Systems Management*. 2nd ed. Chichester, West Sussex, United Kingdom: John Wiley & Sons.

Harel, D., and B. Rumpe. 2004. "Meaningful Modeling: What's the Semantics of "Semantics"?". *Computer* 37(10):64–72. DOI: 10.1109/MC.2004.172.

Juan, A. A., and M. Rabe. 2013. "Combining Simulation with Heuristics to Solve Stochastic Routing and Scheduling Problems". In *Proceedings of the ASIM 2013 Conference on Simulation in Production and Logistics*, edited by W. Dangelmaier et al., 641–649. Paderborn, Germany: Heinz-Nixdorf-Institut.

Kengpol, A. 2008. "Design of a Decision Support System to Evaluate Logistics Distribution Network in Greater Mekong Subregion Countries". *International Journal of Production Economics* 115(2):388–399. DOI: 10.1016/j.ijpe.2007.10.025.

Kieburtz, R. B., L. McKinney, J. M. Bell, J. Hook, A. Kotov, J. Lewis, D. P. Oliva, T. Sheard, I. Smith, and L. Walton. 1996. "A Software Engineering Experiment in Software Component Generation". In *Proceedings of the 18th International Conference on Software Engineering*, March 25th-29th, Berlin, Germany, 542–552.

Kuhn, A., B. Hellingrath, and H. Hinrichs. 2008. „Logistische Assistenzsysteme". In *Software in der Logistik. Weltweit sichere Supply Chains*, edited by M. ten Hompel, 20–26. München, Germany: Huss (Logistik Praxis).

Liebler, K., U. Beissert, M. Motta, and A. Wagenitz. 2013. "Introduction OTD-Net and LAS: Order-to-delivery Network Simulation and Decision Support Systems in Complex Production and Logistics Networks". In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy et al., 439–451. Piscataway, New Jersey: IEEE.

Mnih, V., K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. 2013. "Playing Atari with Deep Reinforcement Learning". In *NIPS 2013 Deep Learning Workshop*, December 9th, Lake Tahoe, USA.

Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. 2015. "Human-level Control through Deep Reinforcement Learning". In *Nature* 518(7540):529–533. DOI: 10.1038/nature14236.

Object Management Group. 2017. *Unified Modeling Language Specification*. http://www.omg.org/spec/UML/2.5.1/PDF/, accessed March 29th, 2018.

Pétrowski, A., and S. Ben-Hamida. 2017. *Evolutionary Algorithms*. London: ISTE.

Rabe, M., F. Dross, and A. Wuttke. 2017a. "Combining a Discrete-event Simulation Model of a Logistics Network with Deep Reinforcement Learning". In *Proceedings of the MIC and MAEB 2017 Conference*, July 4th-7th, Barcelona, Spain, 765–774.

Rabe, M., F. Dross, D. Schmitt, M. Ammouriova, and C. Ipsen. 2017b. "Decision Support for Logistics Networks in Materials Trading Using a Simheuristic Framework and User-generated Action Types". In *Proceedings of the ASIM 2017 Conference*, edited by S. Wenzel and T. Peter, 109–118. Kassel, Germany: Kassel University Press.

Rabe, M., D. Schmitt, and F. Dross. 2017c. "Method to Model Actions for Discrete-event Simulations of Logistics Networks". In *Proceedings of the 2017 Winter Simulation Conference*, edited by W.K.V. Chan et al., 3370–3381. Piscataway, New Jersey: IEEE.

Rabe, M., D. Schmitt, and M. Ammouriova. 2018. "Utilizing Domain-specific Information in Decision Support for Logistics Networks". In *Proceedings of the International Conference on Dynamics in Logistics (LDIC) 2018*, edited by M. Freitag et al., 413–417. Cham, Switzerland: Springer.

Rushton, A., P. Croucher, and P. Baker. 2006. *The Handbook of Logistics and Distribution Management*. 3rd ed. London: Kogan Page.

Spears, W. M. 2000. *Evolutionary Algorithms: The Role of Mutation and Recombination*. Berlin: Springer.

Stockheim, T., M. Schwind, and W. Koenig. 2003. "A Reinforcement Learning Approach for Supply Chain Management". In *1st European Workshop on Multi-Agent Systems*, December 18th – 19th, Oxford, UK.

Sutton, R. S., and A. G. Barto. 1998. *Reinforcement Learning: An introduction*. Cambridge: MIT press.

Van Deursen, A., and P. Klint. 1998. "Little Languages: Little Maintenance?". *Journal of Software Maintenance* 10(2):75–92.

Van Deursen, A., P. Klint, and J. Visser. 2000. "Domain-specific Languages: An Annotated Bibliography". *ACM Sigplan Notices* 35(6):26–36. DOI: 10.1145/352029.352035.

AUTHOR BIOGRAPHIES

MARKUS RABE is full professor for IT in Production and Logistics (ITPL) at the Technical University Dortmund. Until 2010 he had been with Fraunhofer IPK in Berlin as head of the corporate logistics and processes department, head of the central IT department, and a member of the institute direction circle. His research focus is on information systems for supply chains, production planning, and simulation. Markus Rabe is vice chair of the "Simulation in Production and Logistics" group of the simulation society ASIM, member of the advisory board of the Journal of Simulation, member of several conference program committees, has chaired the ASIM SPL conference in 1998, 2000, 2004, 2008, and 2015, and was local chair of the WSC'2012 in Berlin. More than 190 publications and editions report from his work. His e-mail address is markus.rabe@tu-dortmund.de.

MAJSA AMMOURIOVA is a researcher at ITPL at the Technical University Dortmund. She got her master's degree in industrial engineering from the University of Jordan. Since 2016, her research focuses on increasing the performance of a logistics assistance system for distribution networks by investigating heuristics implemented in a simheuristic framework. Her email address is majsa.ammouriova@tu-dortmund.de.

DOMINIK SCHMITT works as a researcher at the Graduate School of Logistics and the ITPL at the Technical University Dortmund. He holds a diploma in Computer Science from the Technical University Dortmund. He graduated with a diploma thesis on the development of a computer-based model for the representation of transformable production systems for scheduling algorithms. Since 2016 he focuses his research on how to make the modeling of changes in discrete-event simulation models more accessible for non-simulation experts. His e-mail address is dominik.schmitt@tu-dortmund.de.