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

IMPROVING THE PERFORMANCE OF A LOGISTICS ASSISTANCE SYSTEM FOR MATERIALS TRADING NETWORKS BY GROUPING SIMILAR ACTIONS

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

Decision makers (DMs) for logistics networks (LNWs) have the complex task of maintaining their networks in good conditions while internal and external demands are changing. Therefore, the DMs need to identify promising actions in order to adapt to the LNW's changing state, e.g., increasing the stock level of stock keeping units (SKUs). The authors have developed a logistics assistance system (LAS) that automatically alters the LNW's model, for improving it under changing conditions, by applying actions and evaluating their effects on the LNW's performance. Promising actions are suggested to the DM. As the LNW grows in size, the number of potential actions increases and therefore, the response time of the LAS increases as well under the additional computational burden. In this paper, the authors describe a novel concept for reducing the number of actions by grouping similar actions together, leading to faster convergence and shorter response time of the LAS.

1 INTRODUCTION

Decision makers aim to improve the performance of their logistics networks. The performance is represented by key performance indicators (KPIs), e.g., the costs of the logistics network and service level (Rushton et al. 2006; Brandimarte and Zotteri 2007). The logistics network's costs include all costs associated with logistics activities, such as activities related to the supply of products, their handling in the logistics network's sites, and their distribution to the customers (Ghiani et al. 2013). Service level indicates the overall degree of customer satisfaction, such as on-time delivery, product characteristics, and price (Ghiani et al. 2013). Decision makers address actions to be applied, and hence, may affect the logistics network's KPIs. An example of an action is a centralization of an SKU in a site. This action affects the storage and the supply of this specific SKU in the specified site and in other sites of the logistics network as well. Actions affect various logistics network activities, which may lead to contrary effects on several KPIs (Rushton et al. 2006). For instance, a decentralization of an SKU may increase the service level but might also lead to an increase in the costs of the logistics networks.

In order to support DMs, a logistics assistance system has been developed by Rabe et al. (2017b). The LAS is based on a simheuristic framework, in which a heuristic algorithm and a discrete event simulation (DES) are combined (Juan and Rabe 2013). The heuristic algorithm is exploring potential actions, searching for the most promising ones. The DES is used to evaluate the action's impact on the LNW's performance. These steps, exploration and evaluation, are iterated until promising actions are found. An increased logistics network's size leads to a growing number of actions to be explored. As the number of actions grows, the exploration time increases as well under the additional range of actions and, thus, the total response time of the LAS. To deal with this problem, the authors are investigating a novel approach for grouping similar actions together. These actions are aggregated into a single action, resulting in a reduced number of potential actions to be investigated and, eventually, to a faster LAS's response time.

This paper is organized as follows: Section 2 gives an overview of related work. Section 3 introduces actions and their derivation and grouping. In section 4, experiments and results comparing various actions are described. Section 5 presents a conclusion and an outlook.

2 RELATED WORK

2.1 Simulation Tool SimChain

SimChain is a simulation tool for logistics networks (Gutenschwager and Alicke 2004) that has been used for the simulation part of the authors' LAS. SimChain offers two main components: a data model for logistics networks and a set of generic building blocks. The data model contains over 50 tables storing the complete parameterization for these building blocks, which can be initialized from the data model's data. Thus, the building blocks represent the logistics network's structure and configuration, e.g., suppliers, sites, customers, transport relations, SKUs, transport frequencies, customer demands, SKUs in a site, and their stock level. In this paper, the technical names of SimChain's data model have been adapted to address the application domain of logistics networks and to improve the readability, as shown in Figure 1. At run time, the simulation model is dynamically instantiated from the data model. The working principles of the simulation tool SimChain are shown in Figure 2.

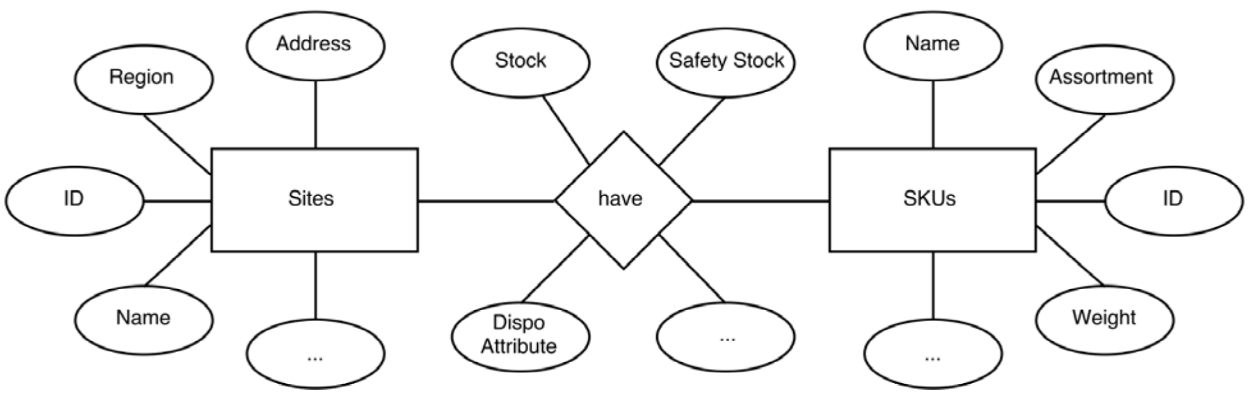

Figure 1: Data Model of SimChain (names of tables and attributes adapted).

2.2 Decision Support for Logistics Networks

Systems in logistics networks for supporting managers in the process of conducting critical decisions are called decision support systems or logistics assistance systems (Blutner et al. 2007; Kuhn et al. 2008). Throughout the literature, the terms logistics assistance system and decision support system in logistics are often used synonymously. According to Kengpol (2008) and Shim et al. (2002), the term decision support system is used more widely in this domain. However, the authors will consistently use the term logistics assistance system, in order to highlight the system's domain. For its targeted critical decision situations, a LAS is supposed to suggest the most promising decision possibilities by evaluating their effects on the logistics system and providing the results to the decision makers.

In the literature, there are several logistics assistance systems using different methods and focusing on various logistics problems. For instance, Liebler et al. (2013) describe a LAS for complex production and logistics networks in the automotive sector. This LAS is using an order-to-delivery network simulation, called OTD-NET. Deiseroth et al. (2008) give a detailed overview of a simulation-based LAS for global supply chains with focus on logistical questions in disposition. In Bockholt et al. (2011), a system for decision support in collaborative supply chain planning is introduced. As in Liebler et al. (2013), both approaches are focused on supply chains in the automotive sector. However, these LAS don't offer the user a possibility of adding or changing potential actions. Therefore, adjusting the corresponding LNW's state is restricted to predefined actions.

Figure 2: Working principle of the simulation tool SimChain (Rabe et al. 2017a).

2.3 Logistics Assistance System for Logistics Networks in Materials Trading

A logistics assistance system for logistics networks in materials trading has been developed by the authors. A simplified version of the LAS's architecture is given in Figure 3. In this section, a brief overview of this system is provided. For a full version of the system's architecture and a more detailed description of its working principles, the reader is kindly referred to previous publications of the authors (Rabe et al. 2017b; Rabe et al. 2018a).

For storing and utilizing data of their logistics networks, companies typically use some kind of basic software, such as SAP R/3. The network's data are automatically extracted, loaded, and transformed into a data model for the simulation tool SimChain. Therefore, the data model is stored in a MySQL database that contains the current network's state. The LAS uses a simheuristic approach (Juan and Rabe 2013). Thus, the performance of the logistics network is evaluated by a data-driven discrete event simulation. In each simulation run, the simulation model is dynamically generated from the database's data. After a simulation run, the experiments' results are forwarded to the heuristic unit (HU). The HU may improve the logistics network's performance by altering its state. Therefore, actions can be applied, e.g., increasing a transport relation's frequency or centralizing an assortment at a site. A detailed description of the HU is provided in section 2.4. Promising actions are determined by the heuristic unit and forwarded to the execution engine (EE). The EE transforms these actions into changes to the underlying database. Based on the altered data in the database, a new simulation model gets instantiated. In order to evaluate the action's effect on the LNW's performance, a simulation run is initiated. The simulation results are transferred back to the HU, which may manipulate the logistics network's state for further improvements. This process runs iteratively until a certain termination criterion is reached, e.g., a distinct number of iterations or a specific quality of the logistics network's performance. When terminating, the suggested actions are provided to the decision maker.

In order to improve the LAS's flexibility, users are able to create new action types (AT) or alter existing ones. An action type is a generalization of similar actions. An action type may describe the increase of stock level of any SKU in any site by any value. Corresponding actions specify the action type by adding distinct parameter values, e.g., increasing the stock of SKU A in site S by 10%, with A and S being a distinct SKU and a specific site, respectively. For modeling user-generated action types, the user has access to the action type designer, a textual or graphical interface. From the action type designer, the user can access all existing action types. In addition, the user has access to predefined constructs of a domain-specific modeling language. This modeling language is restricted to the domain of describing actions in logistics networks of materials trading. Thus, the complexity of modeling action types may get reduced, compared to implementing action types' effects on the data model directly, e.g., by writing specific SQL code. Modeled action types are stored in the action type directory.

The action type directory and, therefore, all action types can be accessed from the scenario builder, a graphical user interface. In the scenario builder, the user can select any action types that should be taken into account for optimizing the logistics network's performance. All actions are derived from the selected action types and provided to the heuristic unit for further investigation. These actions define the HU's search space. Additionally, the user may manipulate the logistics network's state directly. Therefore, the user selects an action type and adds its required parameter values in order to derive one or more actions. These actions are forwarded to the simheuristic framework for evaluation, as shown in Figure 3. Subsequently, the action's effects on the performance of the logistics network will be displayed to the user. A more detailed description of the scenario builder and the process of deriving actions is provided in section 3.1.

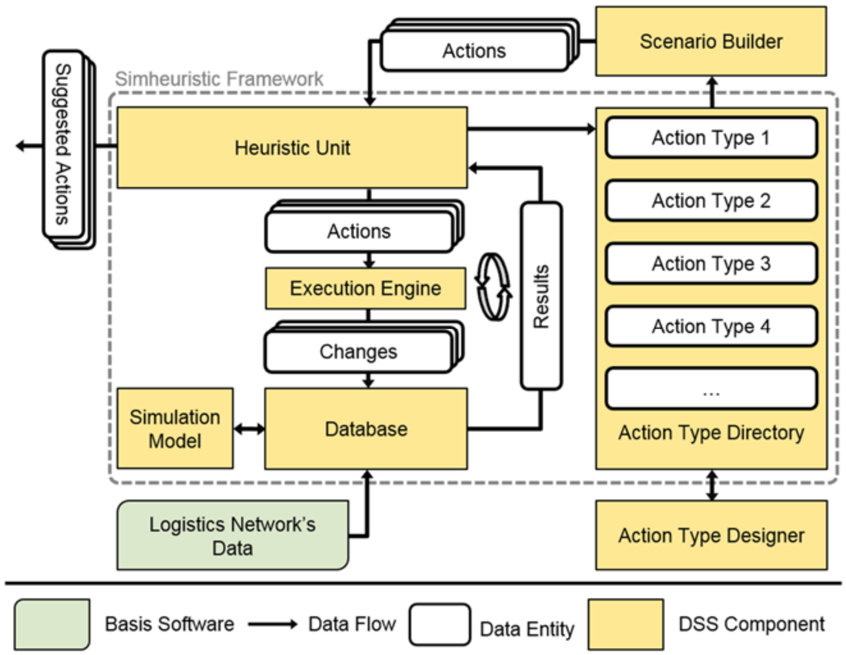

Figure 3: Architecture of the logistics assistance system, based on Rabe et al. (2017b).

2.4 Realization of the LAS's Heuristic Unit using an Evolutionary Algorithm

The LAS's heuristic unit is searching for the most promising action sets, a group of actions, for the current state of the logistics network. For this purpose, the HU is exploring a search space, consisting of the potential actions. 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). An Evolutionary Algorithm (EA) is implemented in the heuristic unit to search for promising action sets, as EA is proposed to solve such type of problems (Pétrowski and Ben-Hamida 2017; Osaba et al. 2014). In addition, EA is often used for evolving most promising solutions under hard CPU time restrictions (Tan et al. 2005). Additionally, a Deep Reinforcement Learning (DRL) algorithm is implemented in the HU. A detailed description of the DRL's implementation is found in Rabe et al. (2017a) and not discussed here.

An *Evolutionary Algorithm* is a stochastic, iterative, population-based search and optimization algorithm mimicking the mechanics of natural evolutionary processes (cf. Ahn 2006; Pétrowski and Ben-Hamida 2017; Bozorg-Haddad et al. 2017; Nissen and Biethahn 1995). Population-based search means that the algorithm explores a population while searching for a candidate solution. It provides good but non-optimal solutions to problems that cannot be solved by exact methods in a reasonable amount of time (Pétrowski and Ben-Hamida 2017), such as combinatorial problems.

The algorithm starts with the generation of an initial population that consists of a number of individuals. An individual represents a solution (cf. Spears 2000; Ahn 2006; Bozorg-Haddad et al. 2017). The solution represents a list of actions to be applied (Rabe et al. 2018a). Actions in a solution are selected randomly from the search space. Then, the individuals are evaluated by assigning a fitness value to each individual. A fitness value assesses the quality of an individual and is used to differentiate between good and bad solutions (Bozorg-Haddad et al. 2017). New individuals in subsequent generations are formed in an evolutionary process that includes operations such as selection, crossover (also called recombination), and mutation (cf. Spears 2000; Tan et al. 2005; Ahn 2006; Bozorg-Haddad et al. 2017; Pétrowski and Ben-Hamida 2017). In the selection operation, individuals are selected based on their quality, hence focusing on the high-fitness individuals. This exploits the available individuals' characteristics and represents the competition between individuals to survive and reproduce (Pétrowski and Ben-Hamida 2017). In crossover, actions of selected mating individuals are exchanged according to uniform distribution. Thus, new individuals are formed, and the current solutions are exploited with the target to find better ones (Ahn 2006). On the other side, mutation perturbs individuals slightly by changing an action in an individual arbitrary. This action is replaced by another action that is selected randomly from the search space. Mutation helps in leaving local optima and, thus, has greater exploratory power than crossover. Therefore, crossover and mutation provide mechanics for exploitation and exploration (Spears 2000). The newly generated individuals are evaluated. Afterwards, the evolution and evaluation of new generations continue iteratively until certain termination criteria are fulfilled. For instance, the algorithm may terminate once a specified number of generations has been generated or the best so far discovered solution has not been changed for a distinct number of generations. A priority-based action's selection utilizing domain-specific information has been investigated and the algorithm's initialization and evolutionary processes have been adapted to utilize these additional information (Rabe et al. 2018a; Rabe et al. 2018b). Accordingly, the initial generation included better solutions, and the algorithm proceeded and found more promising solutions at the termination.

3 ACTIONS IN LOGISTICS NETWORKS FOR MATERIALS TRADING

3.1 The Process of Deriving Actions from Action Types

Actions can be derived from an action type by adding specific values for each input parameter, e.g., for an action type "increase stock" a specific SKU, a specific site, and a specific value for the amount are required. These input parameters can be classified into two types. The first type of parameters are primitive ones, e.g., booleans, integers or strings, such as the additional stock when applying an action "increase stock". The second type of parameters are entities of the logistics network, such as SKUs, sites, suppliers, or transport relations. For instance, in addition to an amount, the action type "increase stock" requires a distinct SKU and a certain site as input values.

When deriving an action from an action type, specific values must be defined for each primitive parameter, such as 10, 20, "true", "false", "Monday", or "Friday". When specifying the affected entities of an action, these entities can be selected individually, e.g., site "Berlin" or transport relation "BerlinToMunich". In contrast to primitive input parameters, entity parameters depend on the current logistics network's state. Selecting a distinct SKU A and site S as input for the action type "increase stock" will have an effect on the logistics network only if site S is storing SKU A, with the stock level being increased by 1, 2… or n.

Additionally, these entities can be selected by certain criteria. Therefore, the authors propose to use a filter construct. A filter construct may filter the affected entities of the logistics network by one or more of their attributes' values (see Figure 1), e.g., all SKUs with a specific id, all sites in a distinct region, or all SKUs that belong to a certain assortment. Those filters may be combined. Therefore, entities can be filtered by multiple filter criteria, e.g., all SKUs of a specific assortment that have a weight of less than 500 kg. When evaluating filter criteria, all corresponding entities will be loaded from the database, e.g., all SKUs. Entities that fulfill all filter criteria will be provided as an input to the action type in the form of a list. Using the concept of filters may reduce the amount of action types in the logistics assistance system. For instance, instead of modeling multiple versions of the same action type that only differ in the affected entities, one basic action type can be modeled and combined with different filter criteria. For example, an action that increases the stock of all SKUs that belong to a specific assortment D in all sites that are assigned to a certain region R by 20 can be derived from the action type "increase stock" by filtering the entities SKUs and sites accordingly and adding 20 as the amount for the increase in stock level. A second action that increases the stock level of a specific SKU A in a distinct site S by 20 can be derived from the same action type by adjusting the filter criteria.

3.2 Use Case for Changing the State of the Logistics Network

Actions can be used to alter the logistics network. For manipulating the network's state, the authors propose the following two use cases:

In the first use case, the user is able to manipulate the logistic network's state directly by deriving and applying one or more actions. Therefore, the user is provided with a graphical interface, the scenario builder, as shown in Figure 3. Through this interface, the user is able to access the available action types. By selecting an action type and specifying its required input, such as specific values or the affected entities, an action can be derived. To reduce the amount of input needed to be specified by the user manually, predefined default values are used. For instance, when deriving an action from the action type "increase stock", the increase of stock level may be set to 10 by default. Thus, the user only needs to define the affected entities, e.g., a specific SKU A in a distinct site S in order to derive an action for increasing the stock level of SKU A in site S by 10. Any logistics network's entity may have a predefined default value assigned to each of its attributes. When deriving an action from an action type, default values can be overwritten by the user. Using predefined default values eases the process of deriving actions by reducing the required input without limiting the system's flexibility. The user may derive any number of actions. These actions are forwarded to the simheuristic framework for evaluation, as described in section 2.3. Consequently, the user can alter the logistic network's state directly and evaluate the effect of these manipulations.

In the second use case, the simheuristic framework is trying to optimize the logistics network's performance automatically. Therefore, the heuristic unit is searching for the most promising actions, which are applied to the database in order to change the logistics network's state. In contrast to the user deriving actions one by one, all potential actions for the current state of the logistics network must be determined in advance. These actions are defining the search space for the heuristic unit. The user may access the scenario builder in order to construct the HU's search space and initiate its search for promising actions. Therefore, the user selects all action types that shall be investigated in the upcoming optimization. Additionally, one or more termination criteria may be defined.

The authors propose to add additional information for each selected action type, e.g., for specifying the action type's input. Otherwise, an unfeasible number of corresponding actions may be derived from a single action type, e.g., actions a_1, a_2, a_3, $\dots a_n$, derived from an action type "increase stock", may increase the stock level of an SKU at a site by 1, 2, 3, …, n, respectively. Whereas, actions b_1, b_2, b_3, $\dots b_m$ may increase the stock by 10 for all SKUs with a weight of more than 100, …, 500, …, m, respectively. Additionally, the affected SKUs and sites could be filtered by any combinations of their attribute's values and logical operators, e.g., all SKUs with a weight of less than 500 kg that belong to an assortment D. Thus, an unfeasible number of actions may be derived leading to an unpractical size of the HU's search space.

The additional information depends on the input parameter's type. For primitive input parameters, default values are predefined and taken into account when deriving actions, as described in use case 1. When configurating the scenario, these values can be either altered by the user or used unchanged. If an action type requires one or more entities as input parameters, these entities must be specified further. For instance, given a logistics network with one site storing three SKUs (SKU1, SKU2, SKU3), whereby SKU1 and SKU2 are assigned to assortment1 and SKU3 belongs to assortment2. When deriving actions from the action type "increase stock", the affected SKUs can be grouped together by one or more of their attributes, e.g., by their id or their attribute "assortment". Grouping the SKUs by their id results in three actions A1, A2, A3 for increasing the stock level of SKU1, SKU2 and SKU3, respectively. When grouping the SKUs by their attribute “assortment”, only two actions are derived: One action for increasing the stock of all SKUs that belong to assortment1 (SKU1 and SKU2), whereas the second action increases the stock of all SKUs that are assigned to assortment2 (SKU3).

Thus, entities can be grouped together by one or more of their attributes, e.g., attribute “assortment” and filtered by one or more of their attributes’ values, e.g., a specific assortment D. When deriving actions from an action type, multiple different values or group conditions for the action type’s input parameters can be used. For instance, when using the action type “increase stock”, the affected SKUs can be grouped by their assortment, the sites by their id and the increase in stock can be set to 10. Thus, a corresponding action will increase the stock level of all SKUs that belong to the same assortment in a site by 10. Another option is to group the SKUs by their id, the sites by their region and the increase of stock level is set to 20. A corresponding action would change the stock of an SKU in all sites that are assigned to the same region by 20. As a result, the amount of actions and, therefore, the size of the search space may be reduced without diminishing the systems flexibility.

In order to reduce the complexity of configuring a scenario, the attributes for grouping entities affected by an action type can be predefined. These default attributes for grouping may be added to the action type during its modeling process. Default attributes for grouping can be overwritten by the users when configuring a scenario.

4 COMPARISON OF ACTION TYPES

The number of actions derived from an action type mainly depends on the logistics network’s state and the attributes by which the required entities are grouped. When deriving an action that affects SKUs, e.g., for increasing the stock level of SKUs in a site, grouping the SKUs by their id typically results in more actions than grouping the SKUs by any other attribute, e.g., assortment. On the other hand, grouping entities by their id, e.g., centralizing an SKU in a site, may enable a more detailed manipulation of the logistics network’s state, because any entity can be used as an input individually. Whereas, grouping entities by any other attribute may result in actions that affect multiple entities simultaneously. This problem may occur for any action type that requires one or more entities as an input.

As the number of actions grows in size, the search space increases progressively. The larger the search space the more time the heuristic units needs searching for promising actions. Thus, there may be a tradeoff between the possibility of manipulating the logistics network’s state on a very detailed level, resulting in a large number of potential actions and the response time of the LAS. In this chapter, the authors are investigating this tradeoff by running several experiments. In each experiment, a single action type is being taken into account that requires at least one or more SKUs as an input (Table 1). For each experiment, SKUs were either grouped by their attribute “assortment” (assortment-based), or by their id (SKU-based). Grouping SKUs could be done by any of the entities’ attributes. However, for the application in practice, enterprises tend to define an action either for single SKUs or for assortments, in order to have their control more transparent. Therefore, a comparison between SKU-based and assortment-based actions may lead to well applicable decision proposals.

4.1 Investigated Action Types

The action types investigated in this paper are “centralization”, “decentralization”, “increase stock” and “decrease stock” (Table 1).

When deriving an action from the action type “centralization”, one site and one or more SKUs are required as an input. These SKUs will be centralized in the given site. Therefore, the corresponding SKUs will be added to the central warehouse. For the sourcing of the affected SKUs, suitable plain suppliers are selected. If necessary, transport relations between these plain suppliers and the central warehouse are added to the logistics network. Furthermore, one or more additional sites can be added as an optional parameter value. These sites define the central warehouse’s sphere of influence. If additional sites are added as an input, the affected SKUs in those additional sites will be sourced from the central warehouse. Therefore, further transport relations might be required for the internal transport of these SKUs. If necessary, additional transport relations are added to the LNW. Any dispensable transport relations will be subsequently removed from the logistics network.

The action type "decentralization" requires one or more SKUs and one or more sites as an input in order to derive an action. A corresponding action will add all given SKUs to all given sites. Therefore, corresponding SKUs that are not already stored in the affected sites will be added to those sites. Subsequently, the sourcing of all affected SKUs will be changed to suitable plain suppliers. Additional transport relations may be added and transport relations which get obsolete are removed, as described for the action type "centralization".

The action types "increase stock" and "decrease stock" are changing the stock level of one or more SKUs at one or more sites. Therefore, one or more SKUs, one or more sites and a specific value for the change in stock level are required as input. Applying a corresponding action changes the stock for all given SKUs in all given sites by adjusting the current stock level accordingly.

For each of these action types, any other entities (e.g., sites) are grouped by their attribute id.

Table 1: Overview of investigated actions.

Assortment-based actions	*SKU-based actions*
Centralizing assortment	Centralizing SKU
Decentralizing assortment	Decentralizing SKU
Increase stock of an assortment	Increase stock of an SKU
Decrease stock of an assortment	Decrease stock of an SKU

4.2 Designed Experiments

Experiments are designed to compare assortment- and SKU-based actions in their convergence and best-found solution by investigating different action types individually. For conducting the experiments, a small-size distribution network consisting of 30 SKUs, classified into 6 assortments, and five sites has been used.

In these experiments, the EA is used for the search of promising actions. The configuration of the algorithms' parameters has been defined as in Table 2. An individual represents a recommended action set and the individual's size defines the number of actions in the action set. If the SKUs are grouped by their id, the individual's size is defined as ten and set to two for assortment-based SKUs. With this selection – considering the average number of SKUs in an assortment – the average number of affected SKUs by an individual in any of the experiments' run is approximately ten.

Table 2: Evolutionary Algorithm experiments' parameters.

Experiment parameter	*Actions based on assortments*	*Actions based on SKUs*
Individual size [action per individual]	2	10
Generation size [individual]	5, 10	10, 20
Crossover probability	0.7	0.7
Mutation probability	0.3	0.3

An initial generation is constructed by selecting actions from the search space randomly. Then, discrete event simulation is used to evaluate the individuals in the initial generation. For the assessment of the initial solutions, the corresponding total costs and service levels are recorded. The individuals' fitness values are represented by the solution's total cost and service level to be minimized and maximized, respectively. In experiment 1, individuals may consist of an increase or decrease in stock level of one or more SKUs in a site more than once. For example, increasing the stock level of SKUs once will lead to an increase of 10 units, while increasing it twice will lead to an increase of 20 units. On the other side, an SKU or an assortment can be centralized only once in a solution. This is due to the serial execution of the individual's actions. For example, an individual has these two actions: centralize assortment1 in site A and centralize assortment1 in site B. As a result, only the second action will influence the logistics distribution network, since the first action will be undone by the second one. In experiment 2 and 3, individuals with two or more actions on the same SKU or assortment are marked as invalid individuals.

Experiments terminate after 100 generations, with a generation representing an iteration of the optimization algorithm. Each experiment has been run for ten times, and the best so far found solution in an experiment run has been recorded. The same seed number has been used for the same run number in each experiment.

4.3 Results and Discussion

The experiments' results are presented in Table 3. The table shows the total costs and service level of the best so far found solution as an average of ten replicates for each experiment. In addition, the average numbers of simulation runs before convergence have been recorded and presented in Table 3. On average, the SKU-based centralization found better solutions than the assortment-based centralization. In all experiments, SKU-based actions have better performance than assortment-based actions.

Table 3: Overview of experiments' results.

Action type	Generation size	Average costs	Average service level	Average number of evaluations before convergence
Experiment 1				
Adjust stock of an assortment	5	65350.1	84.85	94.5
Adjust stock of an assortment	10	65245.8	84.86	166
Adjust stock of an SKU	10	62979.4	85.20	464
Adjust stock of an SKU	20	62821.1	85.39	1086
Experiment 2				
Decentralizing assortment	5	73118.0	84.76	67.5
Decentralizing assortment	10	73118.0	84.76	117
Decentralizing SKU	10	71570.7	84.95	351
Decentralizing SKU	20	71315.7	84.97	788
Experiment 3				
Centralizing assortment	5	65288.4	84.33	99
Centralizing assortment	10	65235.9	84.01	203
Centralizing SKU	10	63151.3	84.63	567
Centralizing SKU	20	62680.6	84.62	1120

The average total costs of the best found solutions are depictured in Figure 4, 5, and 6. The figures show that assortment-based actions converge in fewer generations. Whereas, the best costs found are higher compared to the ones found by SKU-based actions. Assortment-based actions combine the effect of various SKU-based actions together and, thus, may contradict each other. Hence, assortment-based actions' performance is worse than that of SKU-based actions. On the other side, the number of assortment-based actions is smaller than SKU-based actions. Thus, a smaller search space is defined in the case of grouping actions by their attribute "assortment". This leads to a faster convergence of the algorithm and a smaller number of solution evaluations, simulation runs, due to a smaller generation size. Comparing the average total costs found by individuals in a generation gives the same conclusion. However, the authors prefer to show the best found solution in the figures.

Experiment 3 has been repeated on a larger logistics network consisting of 60 SKUs categorized into 12 assortments, and five sites. The experiment's results are presented in Figure 7. These results confirm the findings of the previous experiments 1, 2, and 3. Using assortment-based action types results in a faster convergence than using SKU-based action types. Whereas, better solutions could be found in experiments with SKU-based action types.

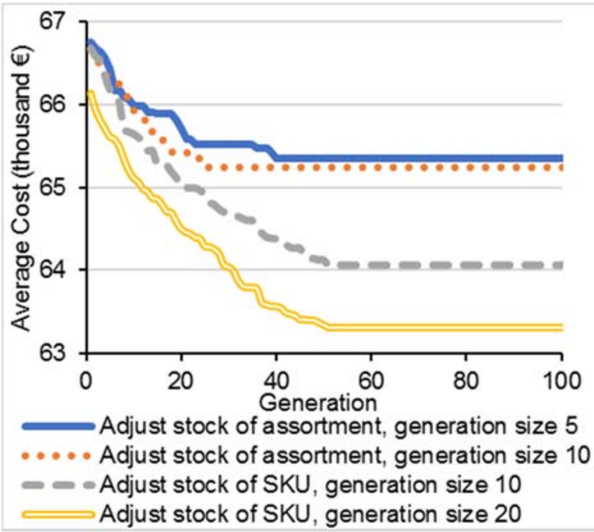

Figure 4: Average best found cost for increase and decrease stock level action type.

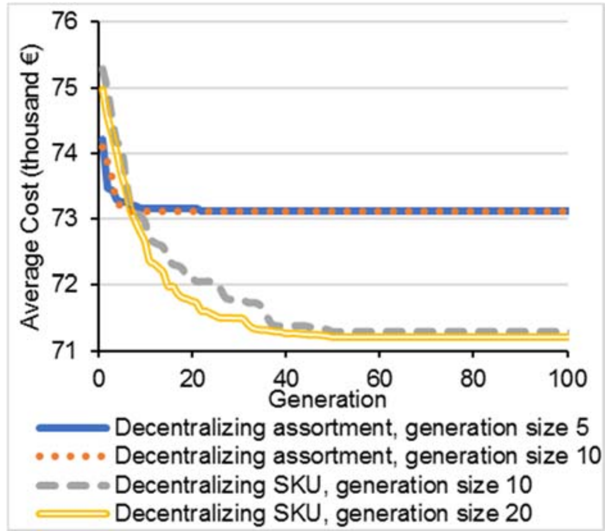

Figure 5: Average best found cost for decentralizing action type.

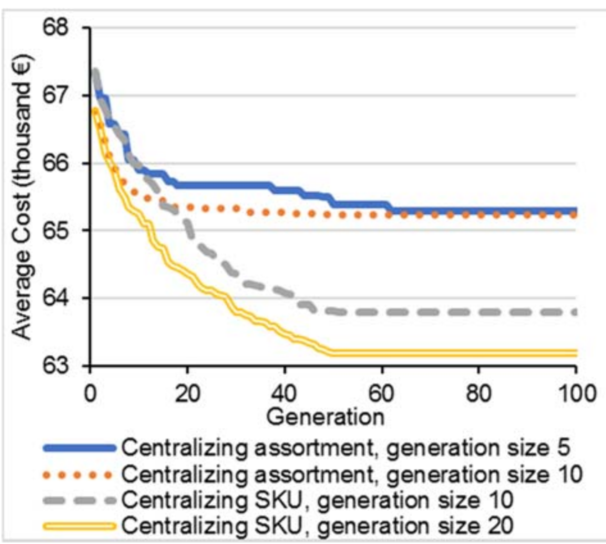

Figure 6: Average best found cost for centralizing action type in 30 SKU logistics network.

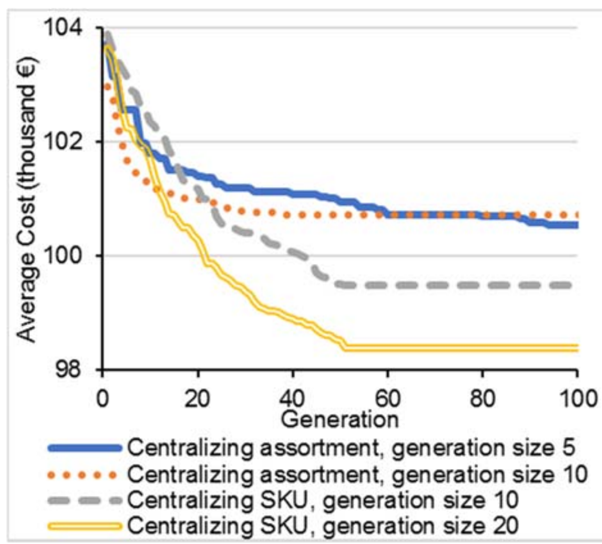

Figure 7: Average best cost for centralizing action type in 60 SKU logistics network.

5 CONCLUSION AND OUTLOOK

With the presented approach, the authors improved the response time of an existing logistics assistance system for logistics networks without significantly diminishing the quality of the suggested solutions. A mechanism for grouping similar actions together into one action has been developed. Grouping may reduce the flexibility of manipulating the logistics network's state compared to applying individual actions on its own. However, grouping may also reduce the search space's size and, therefore, the response time of the logistics assistance system. The authors ran several experiments in order to determine this tradeoff. Results show that the authors' approach reduces the number of required evaluations in the logistics assistance system by approximately 17%, although better solutions can be achieved without using the grouping mechanism. Investigating action sets, consisting of heterogeneous actions, in combination with the grouping mechanism could be a promising candidate for further research.

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.

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.

Gutenschwager, K., and K. Alicke. 2004. "Supply Chain Simulation mit ICON-SimChain". In *Logistik Management*, edited by T. Spengler et al., 161–178. Heidelberg, Germany: Physica.

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.

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

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.

Nissen, V., and J. Biethahn. 1995. "An Introduction to Evolutionary Algorithms". In *Evolutionary Algorithms in Management Applications*, edited by J. Biethahn and V. Nissen, 3–43. Heidelberg: Springer.

Osaba, E., R. Carballedo, F. Diaz, E. Onieva, I. de la Iglesia, and A. Perallos. 2014. "Crossover Versus Mutation: A Comparative Analysis of the Evolutionary Strategy of Genetic Algorithms Applied to Combinatorial Optimization Problems". *The Scientific World Journal* 2014. doi:10.1155/2014/154676.

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.; D. Schmitt, and F. Dross. 2017b. “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. 2018a. “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.

Rabe, M., M. Ammouriova, and D. Schmitt. 2018b. “Utilizing Domain-specific Information for the Optimization of Logistics Networks”. In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe et al. (to be published).

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

Shim, J. P., M. Warkentin, J. F. Courtney, D. J. Power, R. Sharda, and C. Carlsson. 2002. "Past, Present, and Future of Decision Support Technology". *Decision Support Systems* 33(2):111–126.

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

Tan, K. C., E. F. Khor, and T. H. Lee. 2005. *Multiobjective Evolutionary Algorithms and Applications*. New York: Springer.

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.

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.

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.