Simulation in Produktion
und Logistik 2021
Jörg Franke & Peter Schuderer (Hrsg.)
Cuvillier Verlag, Göttingen 2021

Utilising Relations between Actions to Improve the Performance of Optimisation Procedures for Distribution Networks

Nutzung von Aktionsbeziehungen zur Verbesserung der Performance von Optimierungsverfahren für Distributionsnetze

Markus Rabe, TU Dortmund, Dortmund (Germany), markus.rabe@tu-dortmund.de
Majsa Ammouriova, TU Dortmund, Dortmund (Germany), Universitat Oberta de Catalunya, Barcelona (Spain), mammouriova@uoc.edu

Abstract: Optimising distribution networks is challenging because of the complex relationships between entities in the network and their conflicting objectives. It is an NP-hard combinatorial optimisation problem, and a simheuristic approach has been proposed to solve these problems. Simheuristics combines simulation with metaheuristics to optimise a problem. However, in real-world networks, metaheuristics and simulation require long computation time. This paper describes correlation as domain-specific information to guide the search of a metaheuristic algorithm to construct promising solutions in order to reduce the number of simulation runs. The approach was tested on a distribution network of an international trading company. The utilisation of a correlation definition randomly biased the selection of solution elements to construct promising solutions.

1 Introduction

Managing distribution networks is a complex task, and the decision-makers responsible for this task have to select actions from a large number of potential actions to improve the networks' performance. Such actions could, e.g., increase the stock level of a stock keeping unit (SKU) at a site or centralise an SKU at a site in the network. These actions affect either a single entity or multiple entities in the network. Additionally, decision-makers have to consider conflicting goals, such as decreasing costs and increasing the service level (Rushton et al. 2017).

In distribution networks, selected actions are arranged in an action plan. Forming action plans is a combinatorial optimisation problem that is an NP-hard problem in large networks (Dross and Rabe 2014). Thus, decision-makers search for approaches to support them in selecting actions.

Because of distribution network's complexity, a close mathematical formulation of networks cannot be obtained. Instead, simulation is used (Law 2015). A simheuristic approach combining simulation with metaheuristics could optimise these networks (Juan and Rabe 2013). In this approach, simulation evaluates the performance of the network, and metaheuristics optimises it. Even though this approach finds promising solutions in a finite time, its computation time increases with the size of the action space and the complexity of the simulation model. Researchers investigated approaches to improve optimisation approaches, such as simplifying optimisation problems (Ku and Arthanari 2016) and modifying the search mechanism (Grasas et al. 2017).

We defined domain-specific information to guide the search and exploration of actions and, hence, improve the selection of promising actions, such as success and type of changes (Rabe et al. 2017; Rabe et al. 2018a; Rabe et al. 2021). This information is associated with actions and is extracted from the problem. Additionally, correlation as domain-specific information is defined (Rabe et al. 2021). In this paper, we utilise this definition to modify the selection of actions in an evolutionary algorithm to form action plans. Additionally, we evaluate the effect of utilising the correlation definition to search for promising action plans.

This paper is organised as follows: Section 2 gives an overview of the optimisation methods and enhancement approaches, and Section 3 defines correlation. Modifying the action plan's construction in an evolutionary algorithm and its evaluation are presented in Sections 4 and 5, respectively. Section 6 closes with a conclusion.

2 Related Work

This section presents briefly related work in the optimisation of distribution networks and the improvement to optimisation methods.

2.1 Optimisation of Logistics Distribution Networks

Various optimisation methods could be used to optimise logistics distribution networks. Most of the problems in these networks are combinatorial optimisation problems, in which elements are selected from a finite set of elements (Korte and Vygen 2018), such as the travelling salesman problem and vehicle routing problem. These problems are NP-hard problems that are difficult to be solved.

Additionally, distribution networks are characterised by high uncertainty, a large number of entities, e.g., SKUs and sites, and conflicting performance measures (Rushton et al. 2017). A performance measure evaluates the performance of the network (Rushton et al. 2017). Decision-makers can use these measures to compare a network's performance to previous periods or analyse the impact of changes on the network. Examples of conflicting performance measures are costs to be decreased and service level to be increased. These complex networks are challenging to be formulated using mathematical equations and optimised using exact methods.

Researchers used several approaches to optimise these networks. For example, they used metaheuristics and simheuristics (Calvet et al. 2019). In a simheuristic approach, simulation is integrated with metaheuristics. A simulation evaluates changes applied on a network model, and metaheuristics explores the search space for solutions to optimise the network, e.g., the evolutionary algorithm (cf. Talbi (2009) for the description of evolutionary algorithms). These approaches are iterative and terminate if a pre-defined condition is met.

Additionally, researchers have built decision support systems based on optimisation methods. If the system is designed for a logistics network, it is called a logistics assistance system (LAS) (Liebler et al. 2013). Such a system was developed by Dross and Rabe (2014). Figure 1 demonstrates a simplified architecture of the developed LAS. The architecture presents a simheuristic framework that is the base of this LAS to recommend action plans. Discrete event simulation is integrated within a metaheuristic algorithm in the heuristic unit (cf. Fig. 1). This metaheuristic algorithm selects actions and forms action plans, while the simulation evaluates these action plans with respect to selected performance measures, such as costs and service level. The search space consists of actions derived from action types that present a generic description of actions, such as "centralise" without specifying the affected entities.

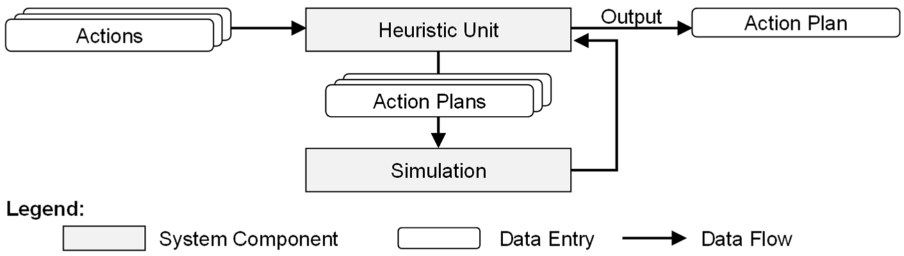

Figure 1: *The simplified architecture of the LAS based on Rabe et al. (2017)*

2.2 Improving the Performance of Optimisation Algorithms

Researchers investigated approaches to improve the performance of optimisation methods, such as screening solutions (Karimi et al. 2017; Bode et al. 2019) and simplifying optimisation problems (Ku and Arthanari 2016). Karimi et al. (2017) clustered solutions for a genetic algorithm at the beginning of the search and showed that their approach outperformed the ordinary genetic algorithm. Bode et al. (2019) defined criteria to screen solutions to reduce the size of the search space. Ku and Arthanari (2016) and Rabe et al. (2018b) replaced the search space of a problem with a smaller search space for easier exploration. Juan et al. (2015) presented a methodology to solve stochastic optimisation problems based on the deterministic version of the problem.

Other researchers focused on modifying the search of the optimisation algorithms. For example, Mane and Narsingrao (2021) used a chaotic sequence to modify the search of a metaheuristic algorithm to solve multi-objective optimisation problems. Alsheddy et al. (2018) integrated a penalty term in their objective function, which is updated during the search to escape local optima. Grasas et al. (2017) proposed a random biased selection of customers in vehicle routing problems to obtain promising solutions.

In addition, researchers investigated hybrid approaches in the optimisation of problems. For example, Li et al. (2020) utilised simulated annealing in modifying solutions in a genetic algorithm. They used their approach to reduce the distribution costs of fresh food logistics. Umetani (2017) used data mining techniques to reduce the size of a search space.

3 Correlation between Actions in the Logistics Assistance System

Correlation in this research is domain-specific information. In this context, we define it as a relation between two sequential actions, e.g., a_i and a_j, and their impact on the performance of a distribution network, $R([a_i, a_j])$. This performance could be measured based on any of the performance measures, such as costs. The relation is defined by comparing $R([a_i, a_j])$ to the estimated impact of the single actions. This estimation assumes that the actions' impact is independent. Hence, their impact together is calculated as $R(a_i) + R(a_j)$. The impact of the actions could be determined based on expert knowledge or a simulation study.

According to the comparison between $R([a_i, a_j])$ and $R(a_i) + R(a_j)$, a positive, a negative, or no recognizable relation could be found. A positive relation $(+)$ exists if the actual impact of the sequential actions exceeds the estimated impact, $R([a_i, a_j]) > R(a_i) + R(a_j)$. In this relation, applying the sequential actions improves the performance of the network. For example, applying a_1 followed by a_2, $[a_1, a_2]$, reduces the costs by 80 € in the example shown in Figure 2. The estimated reduction in costs would have been 65 €, resulting from summing up the reduction caused by a_1 (25 €) and a_2 (40 €).

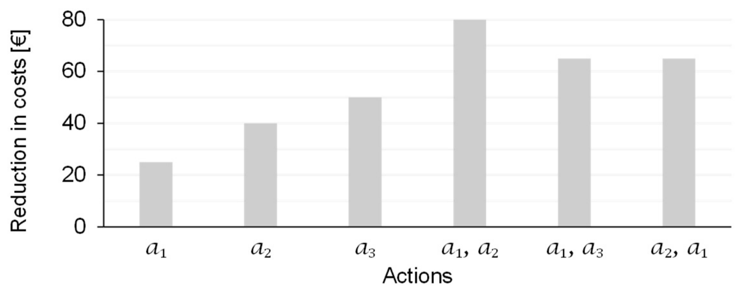

Figure 2: *An impact of actions on the reduction of costs in a distribution network*

In contrast, a negative relation $(-)$ indicates that applying the sequential actions deteriorate the performance of the network, $R([a_i, a_j]) < R(a_i) + R(a_j)$. For example, applying $[a_1, a_3]$ reduces costs by 65 € compared to the estimated reduction of 75 € (Fig. 2). The last relation is defined if the actual impact is very near to the estimated impact of the sequential actions, $R([a_i, a_j]) \approx R(a_i) + R(a_j)$. In stochastic networks, several simulation runs are used to calculate the impact of the actions. Thus, we call this relation a weak relation $(\sim)$, because no significant evidence is available to classify the relationship as positive or negative.

In the LAS, the actions are derived from action types (AT). Thus, relations between action types should be defined. These relation definitions are then assigned to the derived actions. In our work, we assume that actions that affect different entities in a network are independent; thus, the relation between these actions and their impact on the network is weak. Based on this assumption, relations between actions affecting the same entities should be investigated to define the relation between action types.

For example, to define the relation between applying actions from AT_1 followed by actions from AT_2, we investigate $[a_i, a_j]$, where a_i and a_j affect the same entities and are derived from AT_1 and AT_2, respectively. A significant relation defines the impact of applying actions from AT_1 followed by actions from AT_2.

The relations are arranged in a matrix, as shown in Table 1. Rows represent the first applied actions, and columns represent the second applied actions. Thus, applying actions derived from AT_3 followed by actions derived from AT_1 has a negative impact if the actions affect the same entities. For example, action a_3 derived from AT_3 and action a_1 derived from AT_1 affect SKU 1 at site A. Then, the sequence $[a_3, a_1]$ has a negative impact on the performance of the network, but the sequence $[a_1, a_3]$ has a positive impact (cf. Tab. 1). If a_3 affects SKU 3 at site A, both sequences have a weak impact on the performance of the network.

Table 1: *Correlation matrix of action types in the LAS*

	AT_1	AT_2	AT_3	AT_4
AT_1	+	+	+	~
AT_2	+	+	~	−
AT_3	−	−	~	−
AT_4	−	−	~	−

4 Utilising Correlation in a Logistics Assistance System

Once the relations are defined, they are used to bias the selection of actions in constructing action plans in the LAS. The construction of action plans occurs at the initial generation of the evolutionary algorithm. Then, action plans are modified in subsequent generations in the evolutionary algorithm using crossover and mutation.

4.1 Constructing Action Plans Utilising Correlation

Action plans are constructed by selecting actions and adding them sequentially. For example, action plan $[a_3, a_5, a_{10}, a_1]$ is constructed by selecting a_3 as the first action. Then, a_5 is selected and added to the action plan followed by a_{10} and a_1. To incorporate the correlation in constructing action plans, we defined several approaches to alter the selection of actions – three out of five approaches (Rabe et al. 2021) are described here and are utilised in experiments in Section 5. In these approaches, actions are classified into three classes: Positively (A^+), negatively (A^-), and weakly ($A^\sim$) correlated actions. Each class is assigned a probability to select actions from; the highest probability is assigned to A^+ and the lowest to A^-. After selecting a class, an action is selected randomly from the respective class.

In approach (1), the last-added action to an action plan classifies actions into the three classes. For example, after selecting the first action, a_{10}, actions are classified into $A^+_{a_{10}}$, $A^-_{a_{10}}$, and $A^\sim_{a_{10}}$ according to their correlation with a_{10}. Next, a class is selected, and an action is selected from the class. This classification and selection are repeated until the action plan is constructed.

The classification classes are associated with the action plan being constructed in approaches (2) and (3). Thus, after the selection of the first action, the classification classes become A_s^+, $A_s^\sim$, and A_s^- and are updated with each selection of an action. In approach (2), actions are moved from their class to a higher related class if they are classified positively correlated with the newly selected action. Actions are moved from A_s^- to $A_s^\sim$ or from $A_s^\sim$ to A_s^+ if a newly selected action a^* classifies them positive, $A_{a^*}^+$. The actions are moved in the other direction if they are classified negatively by a^*. Figure 3 shows an example. On the one hand, actions a_5 and a_2 are classified positively correlated with a_4 and are moved from their classification class to $A_s^\sim$ and A_s^+, respectively. On the other hand, action a_7 was moved from $A_s^\sim$ to A_s^-.

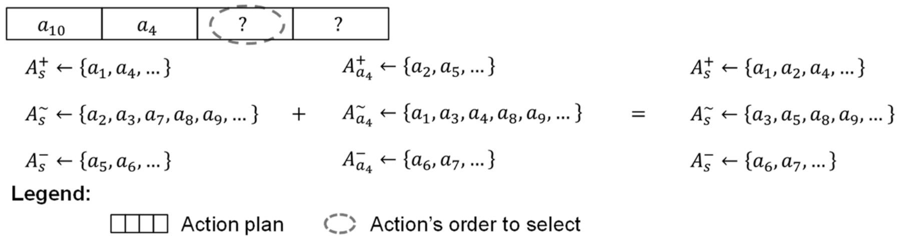

Figure 3: *Updating classification classes based on approach (2)*

In approach (3), if an action is classified as $A_{a^*}^-$ by a selected action a^*, it remains in A_s^-. Thus, once an action is classified with a negative relation with an action in the action plan, it stays negatively related to the whole action plan. An action that is classified positively by a^* is moved from $A_s^\sim$ to A_s^+. If approach (3) is applied in Figure 3, the resulted classification classes A_s^+, $A_s^\sim$, and A_s^- become $\{a_1, a_2, a_4, \dots\}$, $\{a_3, a_8, a_9, \dots\}$, and $\{a_5, a_6, a_7, \dots\}$, respectively. Actions a_5 and a_7 are classified as A_s^-, because they have a negative relation with an action added to the action plan.

4.2 Modifying Action Plans Utilising Correlation

Variation operators modify the action plans in subsequent generations in the evolutionary algorithm, such as crossover and mutation operators. These operators cause destructive variations to the sequence of actions constructed in previous generations to produce new action plans (solutions).

Crossover could take several forms, such as one-point, two-point, and uniform crossover. Since approaches (2) and (3) classify actions according to all previously added actions in an action plan, two-point and uniform crossover do not cause destructive changes in the sequence as in the case of approach (1).

In mutation, one of the actions is replaced by another action from the available actions. The action to be replaced is selected randomly, and the replacement action is selected according to the approaches. For example, if approach (1) is utilised, the replacement action is selected according to the classification based on the preceding action. If approaches (2) or (3) are utilised, all actions before the replaced action should be considered to classify the actions. This replacement could take place for one or more actions in an action plan.

5 Experimental Results and Analysis

In order to evaluate the impact of correlation exploitation, we used a database of a distribution network supplied by an international material trading company. The supplied database stores the network data, such as SKUs, sites, and suppliers. This database is used to instantiate the simulation model in the LAS. The experiments and the correlation between action types in the LAS were defined (Section 5.1). Next, experiment results were collected and analysed (Section 5.2).

5.1 Experiment Setup

In the experiments, the evolutionary algorithm parameters were set to be used in the base experiment and the experiment utilising correlation (the biased experiment). The parameters were set as follows: Crossover probability 0.8, mutation probability 0.3, population size 50, and the maximum number of generations 100. Crossover forms varied in the experiment as one-point, two-point, and uniform crossover. The mutation was set as a one-action or multiple-actions mutation in an action plan. The performance of the evolutionary algorithm is measured by the quality of recommended action plans presented as costs (c) and service level, as well as the number of generations needed to stagnate (N_s). In the experiments, four action types were considered, and each experiment was repeated ten times.

Simulation experiments were used to define the correlation of actions and, accordingly, their action types. To determine the correlation between two action types, pairs of two actions derived from them and affect the same entities in the network were considered. The two actions were applied sequentially on a network model, and their effect on the performance was recorded as actual impact, $R([a_i, a_j])$. This impact was compared to the expected impact of single actions, $R(a_i) + R(a_j)$. For each experiment of a sequential pair of action types, the number of improvements and deterioration in the performance of the network was recorded. Table 2 shows that duplicating an action derived from AT_1 reduced costs in 53% of the pairs. Applying action derived from AT_1 followed by an action derived from AT_2 reduced costs in 75% of the actions pairs that affect the same entities.

Table 2: *Sample of changes in costs after applying two sequential actions affecting the same entity*

Action type of first action	Action type of second action	Percentage of action pairs that have		
		Positive impact	No impact	Negative impact
AT_1	AT_1	53	27	20
AT_1	AT_2	75	17	8
AT_1	AT_3	50	17	33

In order to define the impact of the sequential actions, the sign test (cf. Sheskin 2011) was used. In the test, the median of the changes presents this relation; a positive median indicates a positive relation. The test conclusions are presented in the correlation matrix in Table 1. The positive relation is defined for actions derived from AT_1 followed by actions derived from AT_2 with a p-value of 0.0000. The relation was neither significantly positive nor negative for actions derived from AT_4 followed by actions derived from AT_3; hence, the relation is weak (cf. Tab. 1).

5.2 Results and Discussion

First, correlation approaches for selecting actions were compared. The approaches differ significantly with respect to the quality of found solutions (cf. Fig 4). The CD in Figure 4 is the critical difference calculated in the Fishers test (cf. Sheskin 2011) to identify significant factors after an ANOVA analysis. A similar analysis was performed considering the service level. These approaches do not differ significantly with respect to N_s. From these comparisons, we selected approach (3) for further experiments. In this approach, an action classified negative by an action in the action plan stays negative for the action plan.

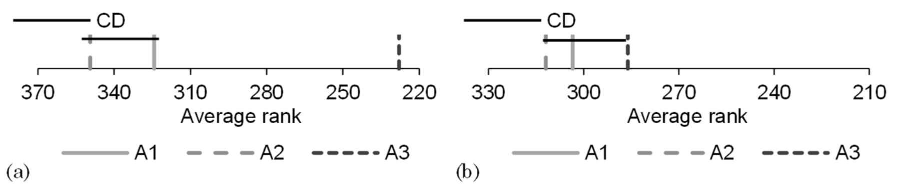

Figure 4: *Critical difference plots for the comparison of correlation approaches: (a) c and (b) N_s*

Next, we analysed the effect of utilising correlation (the biased experiment) compared to the base experiment. In this comparison, approach (3) was utilised in constructing action plans in the initial generation, initial solution. Table 3 summarises this comparison. According to the experiments, utilising correlation led to better solutions with a p-value of 0.000 in the Mann-Whitney U test (cf. Sheskin 2011). However, the base experiments converged earlier.

Table 3: *Comparison between the base experiment and biased experiments for approach (3)*

Crossover	Mutation	Base experiment		Biased experiment		Gap analysis	
		c [€] (1)	N_s (2)	c [€] (3)	N_s (4)	(3) − (1)	(4) − (2)
One-point	One-action	90,420	76	90,835	76	415	0
One-point	Multi-actions	91,414	41	90,802	81	-612	40
Two-point	One-action	90,630	73	90,106	69	-524	-4
Two-point	Multi-actions	90,673	65	89,883	83	-790	18
Uniform	One-action	90,086	89	89,765	73	-321	-17
Uniform	Multi-actions	91,699	63	90,820	74	-879	11

In the second part of the experiment, the effect of utilising correlation in subsequent generations was analysed. Utilising correlation in these generations led to finding promising solutions with a p-value less than 0.1000, in general. To reduce N_s, specific combinations of crossover and mutation were needed.

We defined relations between actions and their impact on the performance of the network. These relations guided the exploration of actions and constructing action plans. As a result, promising solutions were found because actions with a negative impact on the network with respect to other actions get a lower selection probability. Utilising specific knowledge does not support early convergence with all variation operators' combinations. However, finding promising solutions earlier enables us to terminate the search in a fewer number of generations; thus, in a lower number of simulation runs.

6 Conclusion and Outlook

In this paper, we have presented an evaluation of domain-specific information, correlation. Correlation defines the relationship between sequential actions and their impact on the performance of a distribution network. This relation guides the exploration of actions and the construction of action plans to recommend promising action plans that improve the network. Recommending promising solutions in early generations could result in decreasing the number of simulation runs. For further research, correlation definition could be combined with other defined domain-specific information, such as success and type of changes.

References

Alsheddy, A.; Voudouris, C.; Tsang, E. P. K.; Alhindi, A.: Guided Local Search. In: Martí, R.; Pardalos, P. M.; Resende, M. G. (Eds.): Handbook of Heuristics. Cham, Switzerland: Springer, 2018, pp. 261–297.

Bode, F.; Reed, P.; Reuschen, S.; Nowak, W.: Search Space Representation and Reduction Methods to Enhance Multiobjective Water Supply Monitoring Design. Water Resources Research 55 (2019) 3, pp. 2257–2278.

Calvet, L.; Wang, D.; Juan, A. A.; Bové, L.: Solving the Multidepot Vehicle Routing Problem with Limited Depot Capacity and Stochastic Demands. International Transactions in Operational Research 26 (2019) 2, pp. 458–484.

Dross, F.; Rabe, M.: A SimHeuristic Framework as a Decision Support System for Large Logistics Networks with Complex KPIs. In: Wittmann, J.; Deatcu, C. (Eds.): Proceedings of the 22nd Symposium Simulationstechnik, Berlin, Germany, 3–5 September. 2014, pp. 247–254.

Grasas, A.; Juan, A. A.; Faulin, J.; de Armas, J.; Ramalhinho, H.: Biased Randomization of Heuristics Using Skewed Probability Distributions: A Survey and some Applications. Computers and Industrial Engineering 110 (2017) August, pp. 216–228.

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 Production and Logistics – Entscheidungsunterstützung von der Planung bis zur Steuerung. Paderborn, Germany: HNI-Verlagsschriftenreihe, 2013, pp. 641–649.

Juan, A. A.; Faulin, J.; Grasman, S. E.; Rabe, M.; Figueira, G.: A Review of Simheuristics: Extending Metaheuristics to Deal with Stochastic Combinatorial Optimization Problems. Operations Research Perspectives 2 (2015) 1, pp. 62–72.

Karimi, M. B.; Isazadeh, A.; Rahmani, A. M.: QoS-aware Service Composition in Cloud Computing Using Data Mining Techniques and Genetic Algorithm. The Journal of Supercomputing 73 (2017) 4, pp. 1387–1415.

Korte, B.; Vygen, J.: Combinatorial Optimization: Theory and Algorithms. 6th ed. Berlin, Germany: Springer, 2018.

Ku, D.; Arthanari, T. S.: On the Abstraction Method for the Container Relocation Problem. Computers and Operations Research 68 (2016), pp. 110–122.

Law, A. M.: Simulation Modeling and Analysis. 5th ed. New York City, New York: McGraw-Hill, 2015.

Li, D.; Cao, Q.; Zuo, M.; Xu, F.: Optimization of Green Fresh Food Logistics with Heterogeneous Fleet Vehicle Route Problem by Improved Genetic Algorithm. Sustainability 12 (2020) 5, article 1946.

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. R.; Kuhl, M. E. (Eds.): Proceedings of the 2013 Winter Simulation Conference. Piscataway, New Jersey: IEEE, 2013, pp. 439–451.

Mane, S. U.; Narsingrao, M. R.: A Chaotic-based Improved Many-objective Jaya Algorithm for Many-objective Optimization Problems. International Journal of Industrial Engineering Computations 12 (2021) 1, pp. 49–62.

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 Production and Logistics 2017. Kassel, Germany: Kassel University Press, 2017, pp. 109–118.

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

Rabe, M.; Schmitt, D.; Ammouriova, M.: 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, New Jersey: IEEE, 2018b, pp. 2861–2872.

Rabe, M.; Ammouriova, M.; Schmitt, D.; Dross, F.: Simheuristics Approaches for Efficient Decision-Making Support in Materials Trading Networks. Algorithms 14 (2021) 1, article 23.

Rushton, A.; Croucher, P.; Baker, P.: The Handbook of Logistics and Distribution Management. 6th ed. London, United Kingdom: Kogan Page, 2017.

Sheskin, D.: Handbook of Parametric and Nonparametric Statistical Procedures. 5th ed. Boca Raton, Florida: CRC Press, 2011.

Talbi, E.-G.: Metaheuristics: From Design to Implementation. Hoboken, New Jersey: John Wiley and Sons, 2009.

Umetani, S.: Exploiting Variable Associations to Configure Efficient Local Search Algorithms in Large-scale Binary Integer Programs. European Journal of Operational Research 263 (2017) 1, pp. 72–81.