

# LogFarm: An Open Source Graph-based Simulator for Logistics Networks

Alexander Wuttke<sup>1\*</sup>, Joachim Hunker<sup>2</sup>, Markus Rabe<sup>1</sup>

<sup>1</sup>Department of IT in Production and Logistics, TU Dortmund, Leonhard-Euler-Str. 5, 44227 Dortmund, Germany; \*[alexander2.wuttke@tu-dortmund.de](mailto:alexander2.wuttke@tu-dortmund.de)

<sup>2</sup>Fraunhofer Institute for Software and Systems Engineering, Speicherstr. 6, 44147 Dortmund, Germany;

SNE 34(1), 2024, 43-50, DOI: 10.11128/sne.34.sw.10676  
Based on ASIM SPL Simulation Price 2023  
Received: 2024-02-29; Accepted: 2024-03-03  
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna  
ISSN Print 2305-9974, Online 2306-0271, [www.sne-journal.org](http://www.sne-journal.org)

**Abstract.** Logistics networks are complex systems that are often subject to simulation to gain insights. For the computer-aided simulation, simulation tools are used that are also referred to as simulators. In this article, a novel open-source simulator for logistics networks is introduced. The key feature of the tool is its graph-based mode of operation and its usage of simulation models that are stored in graph databases. The simulator is designed to be used in a data farming framework, which means it can be used to generate data along the simulation. The article discusses the general idea and concept behind the authors' graph-based simulator as well as the implementation of an open-source prototype. The prototype is evaluated through the examination of two use cases and a test on the ARGESIM C14 benchmark.

## Introduction

Logistics plays a vital role in modern society. The execution of logistics processes, such as transporting, handling, and storing, is performed in logistics systems that consist of multiple interrelated actors. Thus, logistics systems have a network-like structure and are often referred to as logistics networks (LNs). Due to their network-like structure, modeling and simulating LNs as graphs has proven to be a promising approach [1]. It also simplifies the application of graph-based algorithms, e.g., to solve vehicle routing problems as shown in [2].

In order to keep LNs in an efficient and competitive state, simulation is a proven approach at the planning and operational stage. An increasingly popular trend in simulation is to use a simulation model to systematically generate data, which is referred to as data farming [3]. With this approach, data with higher quality can be obtained compared to observational data in real-world applications, which are prone to, e.g., noise and measuring errors [4].

A possible application is to use the data with a subsequent knowledge discovery process, which is referred to as farming-for-mining and is detailed in [5]. Simulation is usually carried out using simulation tools. There is a wide range of different simulation tools available that can simulate LNs, such as AnyLogistix [6] and Enterprise Dynamics [7]. However, those simulation tools rely on relational data and modeling or only partially make use of graphs.

The authors propose to use simulation tools that embrace graph thinking and graph-based approaches, which includes the modeling of simulation models as graphs and the usage of graph algorithms for the simulation of LNs. In this article, the authors point out relevant fundamentals on graphs in general and on labeled property graphs (LPGs) in particular, which are a special type of graph that can be enriched with additional information. Then, key concepts of the novel simulation tool are discussed including the usage in a data farming framework. Based on these concepts, the implementation of an open-source prototype is shown and assessed on the basis of two use cases and tested on the ARGESIM C14 benchmark [8].

## 1 Related Work

In this section, a concise introduction to the domain of LNs is given and discussed, how simulation as well as data farming are used in this context. Then, key features of graphs and LPGs are highlighted. Also, the mindset and approaches of graph thinking are briefly explained.

### 1.1 Logistics Networks, Simulation, and Data Farming

LNs are complex systems, consisting of a mesh of different actors such as suppliers, sites, customers, and a multitude of different logistics processes [9]. To keep an LN in a good and competitive state, often the complex interrelationships in an LN are of interest [10]. In this context, simulation has established itself as a proven method for the modeling and analysis of such a complex system [11]. Simulation is defined as the “representation of a system with its dynamic processes in an experimentable model to reach findings, which are transferable to reality; in particular, the processes are developed over time” [12, 3]. The use of a simulation model and efficient experiment design to generate vast amounts of simulation result data is called data farming. According to [3, 1], data farming “is a descriptive metaphor that captures the notion of generating data purposefully to maximize the information yield from simulation models”. A procedure model for data farming in a farming-for-mining-framework can be found in [13].

To support a data farming study, simulation tools are used. Simulation tools are a piece of software, consisting of five main components: kernel, data management, user interface, interfaces to external programs, and a predefined modeling world [12, 14]. The modeling world is particularly relevant for the process of building a simulation model. According to [15], the process of building a simulation model is characterized by three consecutive steps: system analysis (resulting in a conceptual model), formalization (resulting in a formal model), and implementation (resulting in an executable model). To guide these steps, a modeling concept is necessary. Established concepts are building blocks (libraries), object-oriented concepts (following the object-oriented paradigm from software development), languages (higher programming languages), and theoretical concepts such as graphs [16].

### 1.2 Graphs and Graph Thinking

Graph-theoretical modeling concepts are relevant for the representation of complex and interrelated LNs. Although graph theory has been around since Leonhard Euler presented a solution to the problem of the seven bridges of Königsberg, graphs have been gaining interest in research and practice in the last decade [17, 18]. Reasons for this development are advancements in database technology, in particular graph databases, which are a type of NoSQL database system [19]. These developments are driven by the challenges that arise from dependency-oriented and heavily interconnected data, in which explicit interrelations among data are of interest [20]. Graphs are mathematical objects, which naturally describe networks. Typical examples include social networks, public transportation systems, and LNs. Numerous papers report from the successful use of graph-based technologies such as graph databases in logistics, for example, in [21].

Mathematically, a graph is defined as a pair  $G = V, E$ , with  $V$  being a finite set of vertices and  $E$  being a finite set of edges. In a logistics context, the term node is used synonymously with vertices. Since the basic structure of nodes and edges cannot represent complex LNs accurately, both can be detailed using properties, in the form of key-value pairs, and labels to identify nodes and edges. This specific type of graph is called an LPG [22], which is a widely used graph model for graph databases, for example, Neo4j [23]. In [24], it is shown how the concept of an LPG can be used to model LNs. An example of an LPG with two nodes, one edge, three labels, and a total of four properties in a logistics application is shown in Figure 1.

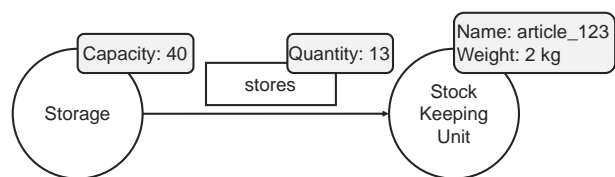


Figure 1: Exemplary labeled property graph.

## 2 The Simulation Tool

In this section, the novel simulation tool LogFarm developed by the authors of this article is introduced. The simulator is designed graph-based and used in a data farming framework.

The simulation subject is targeted to LNs. The introduction of LogFarm consists of three parts. First, the conceptual model of LNs in LogFarm is presented. This defines what part of the real world can be modeled and simulated with the simulation tool. Second, the general software design are described as well as the tool's functionalities. In the last part, a brief presentation of the implemented prototype is given.

### 2.1 Conceptual Model of Logistics Networks

The simulation models that can be interpreted and used by the simulator must follow a predefined form. To define the scope of simulation models on a conceptual level, a conceptual model of generic LNs was derived. It serves as a meta model to describe which aspects of LNs can be simulated and at which level of detail. The conceptual model results from a system analysis (see section 2.1).

The level of detail used for the simulation is set at the level of interaction between facilities of actors in the LNs. The considered types of facilities are suppliers, manufacturers, warehouses, and customers. The physical locations of customers, regardless of being an intermediary or end customer, are also referred to as facilities. The simulation progresses over discrete time intervals, each representing a full day. As a consequence, internal processes of facilities and processes that take less than a full day are abstracted and aggregated. Working hours of facilities can be specified for each day of the week, also allowing for disabling a facility for certain days when the working hours are set to zero.

The simulator is capable of simulating transports between facilities of actors and storage processes that take place in LNs. For the transports, vehicle fleets can be assigned consisting of different types of vehicles that can be specified by the user. If a facility distributes goods to multiple other facilities, a dynamic vehicle routing can be used. The geospatial position of locations is specified through the longitude and latitude. The storage keeping is based on policies that can be chosen from a predefined set. Further simulated processes are production and transformation as well as demand and consumption processes that are necessary to establish a flow of goods. The production processes are the sources and the consumption processes are the sinks of goods. Production and transformation processes can optionally be subject to disruptions, also called malfunctions, that are stochastically triggered. The explicitly simulated entities are facilities, goods and vehicles.

### 2.2 Software Design

The program uses a building-block-oriented modeling concept to create simulation models with regards to the conceptual model, which means that predefined logical entities exist. Each building block resembles a facility of an actor in an LN that acts according to its policy and interacts with other facilities. Therefore, the model can be seen as agent-based.

The building blocks are provided through a building block library that, up to this point, encompasses four different building blocks to model suppliers, manufacturers, warehouses, and customers. Additionally, there is a building block called customer group, which is used to generate large numbers of customers procedurally.

The modeling of a simulation model can take place in the simulation tool itself. LogFarm provides a graphical user interface (GUI) for an editor of simulation models on which, per drag-and-drop, the user can select building blocks from the building block library and place them on a modeling canvas.

Building blocks that should have interactions with other building blocks can also be connected by drag-and-drop of relations. Figure 2 shows the building block library on the left side and on the right side the modeling canvas, where a simple model of two suppliers, one manufacturer, one customer, and three relations can be seen.

Through the GUI of the editor it is also possible to parameterize the buildings blocks after placing them on the modeling canvas. This includes, e.g., setting process rates, defining transformation rules for goods, and enabling malfunctions with certain stochastic traits.

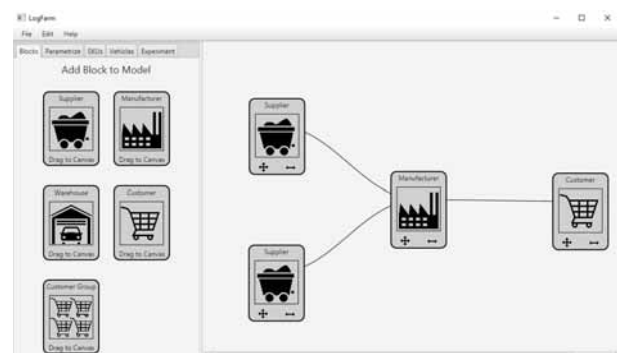
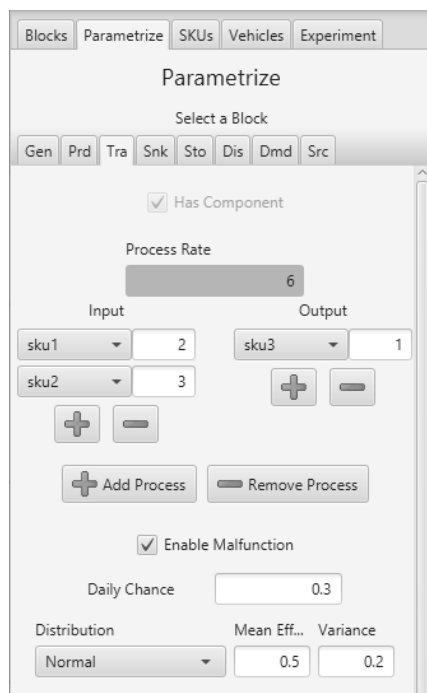
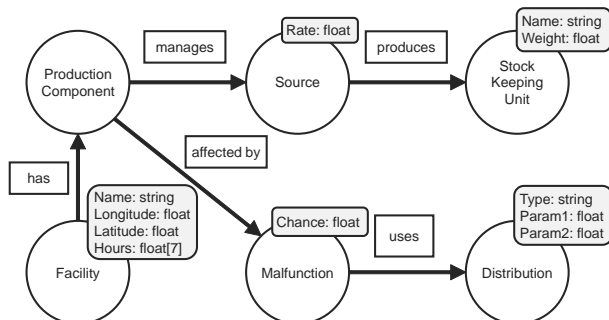


Figure 2: GUI of the editor to build simulation models in the simulation tool.



**Figure 3:** Parametrization window of a block in the simulation tool.



**Figure 4:** Example of a facility with a production component modeled in an LPG.

Besides giving constant values for parameters, it is also possible to let certain parameters be controlled according to an experiment plan, which stems from a design of experiments and is used when multiple simulation runs are conducted. In this case, the value of the parameter is dynamically changed before a simulation run, according to the value of the corresponding factor in the experiment plan for a given simulation run. To link a parameter to a factor, the identification number of the factor must be entered in the text field and it must be left clicked while holding the CTRL-key.

This will also highlight the text field with a yellow background color to make it distinguishable from constant values. An example of the parametrization window is shown in Figure 3.

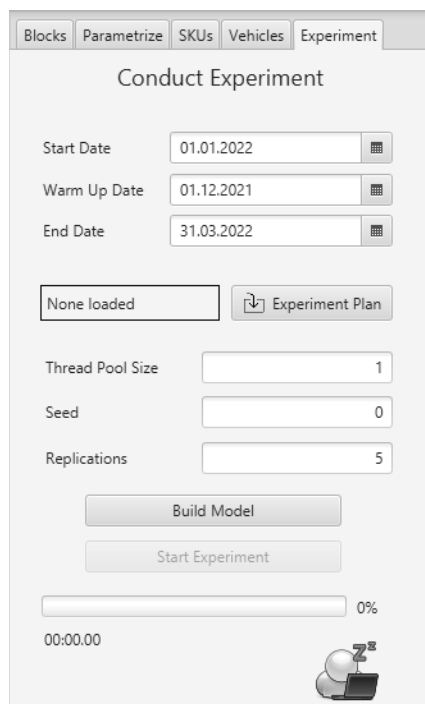
After finishing modeling a simulation model in the editor it can be saved to a graph database in form of an LPG. It is also possible to load a simulation model in the form of an LPG and simulate it.

The modeling of a simulation model as an LPG is guided by a modeling framework that is only briefly described in this article. A more in-depth explanation is provided in prior work of the authors [24]. The modeling framework consists of a set of labels, a set of possible properties, rules for edges, and general modeling rules. In Figure 4, a supplier with its production component affected by malfunctions is modeled, which will serve as an explanatory example. In the case of LogFarm, properties are only attached to nodes. For example, facilities can be detailed by a name, geo-coordinates, and working hours. Those properties correspond to the parameters of the building blocks. Because the facility node is connected to a production component node, the simulator recognizes that the facility is able to execute production processes. What is to be produced is specified by the connected source node and stock keeping unit node. The simulator can also recognize that the production component is affected by malfunctions that occur by a given probability and have a failure duration determined stochastically by a given distribution. The shown example is only a small part of a valid simulation model that can be used for LogFarm. Valid simulation models can feature a much more complex and rich structure of nodes, edges, and properties.

Another functionality of LogFarm is to read externally generated experiment plans for simulation. The experiment plan must be in the CSV file format and can be generated by tools such as MATLAB [25].

Regarding the parameters of the simulation itself, LogFarm supports the specification of a simulation start and end date as well as the definition of a warm-up phase, in which no data are logged or generated. It is also possible to specify a seed to fix the sequence of random numbers and, therefore, enable reproducibility of simulation runs.

Given a simulation model, an experiment plan, and simulation parameters, the actual simulation can take place. The simulation is controlled by an experiment manager, whose task is to execute all simulation runs of the experiment plan.



**Figure 5:** Experiment window of a block in the simulation tool.

The experiment manager of LogFarm is capable of parallel execution of simulation runs. For now, this is limited to the kernels and threads of one local machine, but can be adapted to work in distributed systems. The simulation is carried out with the simulation kernel and yields simulation result data. The GUI to enter simulation parameters and to control the experiment manager is shown in Figure 5.

Usually, simulation is used to gain insights. As described in Section 2.1, simulation can also be used to generate data along the simulation in data farming frameworks. LogFarm is intended to be used in a data farming framework and is, therefore, capable of generating data.

Currently, data on stocks and orders are generated and perpetually written to the database. This enables the detailed analysis of transactional data in the LNs after simulation with, e.g., a farming-for-mining-framework. In the future, more data of interest can be considered and generated. An overview of the software architecture of LogFarm is shown in Figure 6.

## 2.3 Implementation

LogFarm is implemented as an open-source simulator programmed in Java. The source code is hosted on github in the repository found at <https://github.com/Alexw1011/LogFarm>.

The GUI of LogFarm is implemented in JavaFX. For the graph database, Neo4j is used (see Section 2.2). Neo4j is a well-known graph database system that uses an LPG as data model and supports object graph mapping for efficient read and write operations. In Figure 7, a small simulation model for LogFarm in the form of an LPG is visualized by Neo4j.

## 3 Evaluation

To validate the introduced concepts and test the developed prototype, results of two recent case studies are presented. Furthermore, the authors have implemented the ARGESIM C14 benchmarks for comparability.

### 3.1 Case Studies

The prototype was tested on two case studies. One is a typical LN consisting of three suppliers, one manufacturer, one warehouse, and three customers. The other case study is based on a real use case of city logistics on the metropolitan area of Athens, which was examined in a prior study using a different simulation tool [26]. In this case study, a two-echelon LN was examined that featured five suppliers, one consolidation center, and 8,537 customers.

For more information on the case studies, the reader is kindly referred to [27]. The main takeaways from these case studies are that the bottleneck is often not the time for the simulation itself, but the write operations to the graph database, which can be higher than the simulation time. This is especially problematic when using parallel simulation but only one graph database to write to, as a considerable amount of time cannot be used for simulation but is spent to wait for the graph database write operations.

### 3.2 Benchmarks

The following benchmark was conducted accordingly to the ARGESIM C14 benchmark. It is a simplified two-echelon LN with four factories, four distributors, a group of wholesalers, and twelve products.

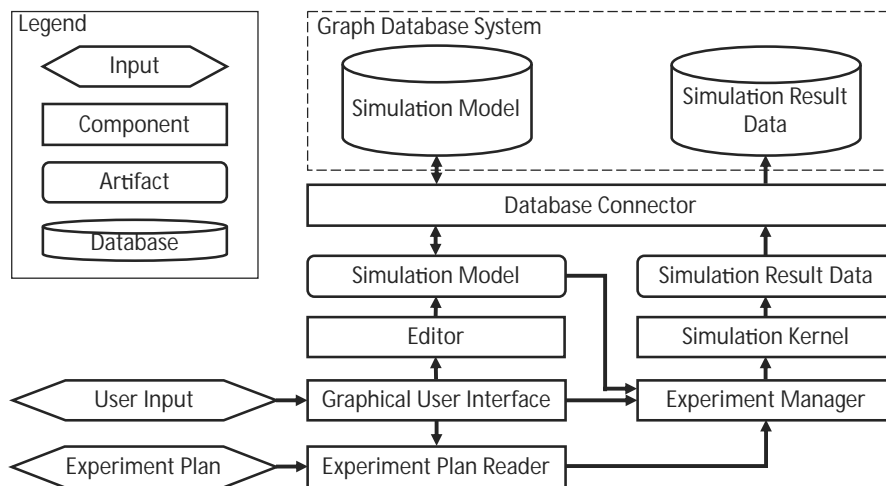


Figure 6: Software architecture of the simulation tool.

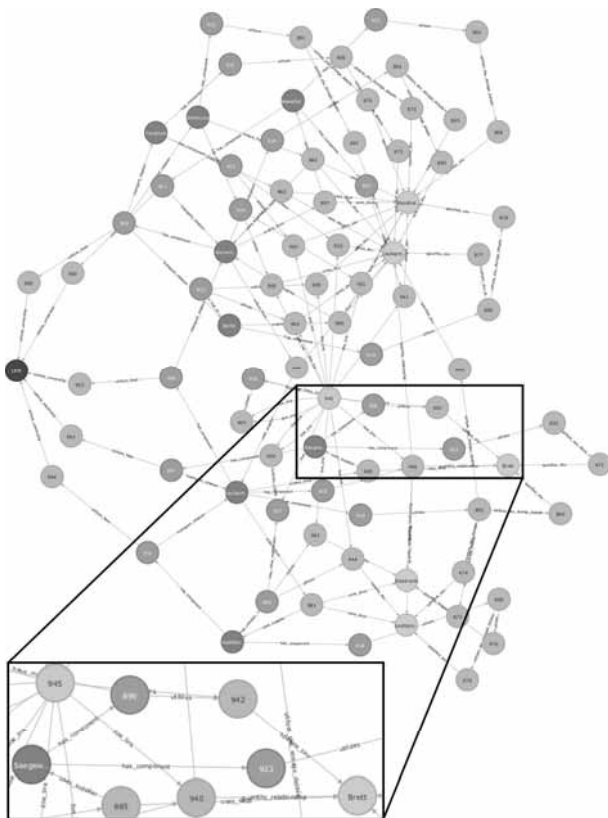


Figure 7: Exemplary simulation model in the Neo4j GUI.

A total of six tasks is considered with different order strategies of the distributors and wholesalers. Following these, the details of the benchmark can be found in [8]. A model has been created and its parameters varied for

the different tasks. To model the factories, a total of four supplier building blocks are used. The four distribution centers are modeled with one warehouse building block each. The group of wholesalers is represented by one customer building block. A screenshot of the model in LogFarm can be seen in Figure 8.

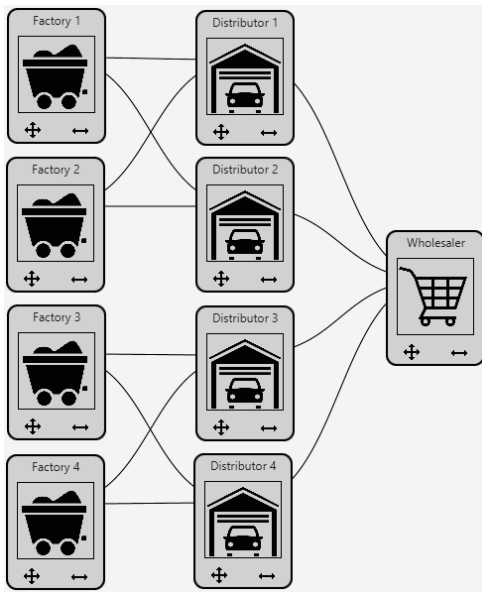
The results are presented in the Tables 1, 2, and 3, as well as Figures 9 and 10. Compared to benchmark results obtained by MATLAB [28] and Enterprise Dynamics [29], the results are similar with only minor deviations, which are likely due to stochastic influences. Based on this comparison, it can be concluded that LogFarm can achieve equivalent results to established simulation tools in the field of LNs.

	C	N	R
<b>min</b>	28,441	190	139.2
<b>max</b>	35,012	253	162.8
<b>mean</b>	32,866	223	148.1
<b>deviation</b>	1,320	12.2	6.1

Table 1: Benchmark results of ARGESIM C14 task a2.

## 4 Conclusion and Outlook

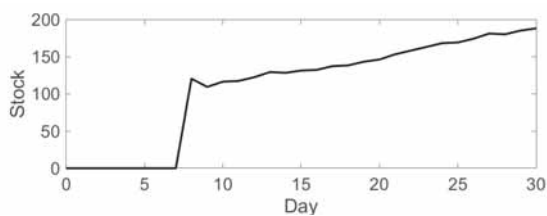
In this paper, an open source graph-based simulator for LNs called LogFarm is presented. In particular, the concept of LPGs is used, which has proven to be an intuitive and appropriate way to model LNs for simulation.



**Figure 8:** Screenshot of the model used for the ARGESIM C14 benchmark.

	C	N	R
<b>min</b>	27,532	193	129.3
<b>max</b>	34,090	253	157.2
<b>mean</b>	32,021	224	143.0
<b>deviation</b>	1,217	12.0	5.8

**Table 2:** Benchmark results of ARGESIM C14 task b2.



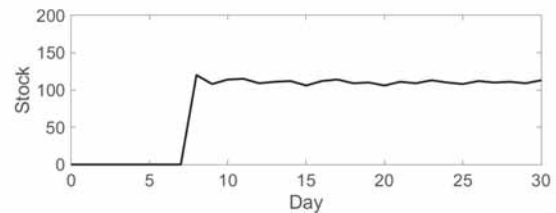
**Figure 9:** Results for the stock in ARGESIM C14 task a1.

Its applicability is shown in two case studies and in the ARGESIM C14 benchmark.

The authors' goal is to stimulate a more extensive use of graph thinking. The introduced simulation tool is a proof of concept, which has shown very promising results and already sparked interest in various new approaches. Although the shown prototype of LogFarm is capable of simulating many typical scenarios of LNs, the range of functions offered is still limited compared

	C	N	R
<b>min</b>	22,882	192	106.9
<b>max</b>	29,980	251	141.5
<b>mean</b>	26,798	224	119.6
<b>deviation</b>	1,122	12.3	5.1

**Table 3:** Benchmark results of ARGESIM C14 task c1.



**Figure 10:** Results for the stock in ARGESIM C14 task b1.

to established simulation tools. Therefore, the next step to increase the maturity level of the software is to add more functionalities like, e.g., distance calculations not only based on geo-coordinates, more options to model stochastic influences, and more realistic ways to generate demand of customers as suggested in [30].

## References

- [1] Domschke W, Drexl A, Klein R, Scholl A. *Einführung in Operations Research*. Wiesbaden, Germany: Springer Gabler. 2015.
- [2] Barceló J, Grzybowska H, Pardo S. Vehicle Routing and Scheduling Models, Simulation and City Logistics: 2: 3. In: *Dynamic Fleet Management: 10*, edited by Zempekis V, Tarantilis CD, Giaglis GM, Minis I, pp. 163–195. Boston, MA, USA: Springer US. 2007.
- [3] Sanchez S. Data Farming: Methodes for the Present, Opportunities for the Future. *ACM Transactions on Modeling and Computer Simulation*. 2020;30(4):1–30.
- [4] García S, Luengo J, Herrera F. *Data Preprocessing in Data Mining*. Cham: Springer International Publishing. 2015.
- [5] Hunker J, Wuttke A, Scheidler AA, Rabe M. A Farming-for-Mining-Framework to Gain Knowledge in Supply Chains. In: *Proceedings of the 2021 Winter Simulation Conference*, edited by Kim S, Feng B, Smith K, Masoud S, Zheng Z, Szabo C, Loper M. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Inc. 2021; pp. 1–12.

- [6] The Anylogic Company. AnyLogistix. 2024.  
URL <https://www.anylogistix.com/>
- [7] InControl. Enterprise Dynamics. 2024.  
URL <https://www.incontrolsim.com/software-platform/>
- [8] Tauböck S. C14 Supply Chain Management - Definition. *SNE Simulation Notes Europe*. 2001; 11(32/33):42–43.
- [9] Christopher M. *Logistics & Supply Chain Management*. Harlow, England: Pearson Education, 5th ed. 2016.
- [10] Pfohl HC. *Logistics Systems: Business Fundamentals*. Wiesbaden, Germany: Springer Gabler. 2022.
- [11] Law A. *Simulation Modeling and Analysis*. NY, USA: McGraw-Hill Education, 5th ed. 2015.
- [12] VDI-Guideline 3633 Part 1. Simulation of Systems in Materials Handling, Logistics and Production: Fundamentals. 2014.
- [13] Hunker J, Scheidler AA, Rabe M, van der Valk H. A New Data Farming Procedure Model for a Farming for Mining Method in Logistics Networks. In: *Proceedings of the 2022 Winter Simulation Conference*, edited by Feng B, Pedrielli G, Peng Y, Shashaani S, Song E, Corlu CG, Lee LH, Chew EP, Roeder T, Lendermann P. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc. 2022; pp. 1461–1472.
- [14] Wenzel S. Simulation logistischer Systeme. In: *Modellierung logistischer Systeme*, edited by Tempelmeier H, pp. 1–34. Wiesbaden, Germany: Springer Gabler. 2018.
- [15] Rabe M, Spieckermann S, Wenzel S. *Verifikation und Validierung für die Simulation in Produktion und Logistik: Vorgehensmodelle und Techniken*. Berlin, Germany: Springer. 2008.
- [16] Gutenschwager K, Rabe M, Spieckermann S, Wenzel S. *Simulation in Produktion und Logistik*. Berlin, Heidelberg, Germany: Springer. 2017.
- [17] Jungnickel D. *Graphs, Networks and Algorithms*. Berlin, Germany: Springer, 4th ed. 2013.
- [18] Robinson I, Webber J, Eifrem E. *Graph Databases: New Opportunities for Connected Data*. Sebastopol, CA: O’Reilly, 2nd ed. 2015.
- [19] Meier A, Kaufmann M. *SQL & NoSQL Databases: Models, Languages, Consistency Options and Architectures for Big Data Management*. Wiesbaden, Germany: Springer Fachmedien. 2019.
- [20] Aggarwal CC. *Data Mining*. Cham: Springer International Publishing. 2015.
- [21] Xiaoliang Z, Qingtao Z, Mingjie T, Hui H. Design and Implementation of Express Logistics System Based on Graph Database and Baidu Map. In: *Proceedings of the 2nd International Conference on Information Technology and Computer Application*, edited by Freris N, Bin Zakaria Z. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Inc. 2020; pp. 396–400.
- [22] Angles R. The Property Graph Database Model. In: *Proceedings of the 12th Alberto Mendelzon International Workshop on Foundations of Data Management*, edited by Olteanu D, Poblete B. Aachen, Germany: CEUR-WS. 2018; pp. 16–26.
- [23] Neo4j. Neo4j. 2024.  
URL <https://neo4j.com/>
- [24] Wuttke A, Hunker J, Rabe M. Modeling of Logistics Networks with Labeled Property Graphs for Simulation in Digital Twins. In: *Proceedings of the EUROSIM Congress 2023*, edited by Mota M, Mendoza A, Scala P. Cham: Springer. 2023; Accepted, to be published.
- [25] MathWorks. MATLAB. 2024.  
URL <https://de.mathworks.com/products/matlab.html>
- [26] Rabe M, Klueter A, Wuttke A. Evaluating the Consolidation of Distribution Flows Using a Discrete Event Supply Chain Simulation Tool: Application to a Case Study in Greece. In: *Proceedings of the 2018 Winter Simulation Conference*, edited by Rabe M, Juan AA, Mustafee A, Skoogh SJ, Johansson B. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc. 2018; pp. 2815–2826.
- [27] Wuttke A. Entwicklung eines graphbasierten Simulators zur Simulation von Logistiknetzwerken in einem Data-Farming-Framework. Master Thesis TU Dortmund, ITPL. 2022.  
URL [https://www.itpl-vw.mb.tu-dortmund.de/publikationen/MA\\_2022\\_Wuttke.pdf](https://www.itpl-vw.mb.tu-dortmund.de/publikationen/MA_2022_Wuttke.pdf)
- [28] Wastian M, Reichl S. An Object-oriented Approach to ARGESIM Benchmark C14 ‘Supply Chain’ using MATLAB. *SNE Simulation Notes Europe*. 2018; 28(1):35–38.
- [29] Gutwenger P, Hoa DD, Puffitsch W, Tauböck S. A Script Language-supported Approach to ARGESIM Benchmark C14 ‘Supply Chain’ in Enterprise Dynamics. *SNE Simulation Notes Europe*. 2011; 21(1):55–56.
- [30] Wuttke A, Hunker J, Scheidler AA, Rabe M. Synthetic Demand Generation with Seasonality for Data Mining on a Data-Farmed Data Basis of a Two-Echelon Supply Chain. *Procedia Computer Science*. 2022;204:226–234.