

LogisticsLab: An academic software for decision-making in logistics

Mike Steglich

University of Applied Sciences Wildau

Hochschulring 1

15745 Wildau, Germany

mike.steglich@th-wildau.de

KEYWORDS

logistical decision-making, transportation problems, network flow problems, traveling salesman problems, Chinese postman problems, vehicle routing planning problems, location problems, problem-based learning and teaching

ABSTRACT

Logistical decision problems are a part of many courses in the field of logistics, management and operations research. It makes sense to illustrate these optimisation problems using case studies, which can be reproduced by students using suitable software. Often, solver add-ins in spreadsheets programs or general optimisation software are used, which on the one hand requires a high level of knowledge in Operations Research and on the other hand does not always allow an intuitive approach. This article describes the academic software LogisticsLab with which the distributors tie in with the idea of interactive decision support systems to systematically combine the experiences and intuitions of human decision-makers with the possibilities of computer-assisted modelling and optimisation of a wide range of logistical decisions.

INTRODUCTION

Students studying logistics management, supply chain management, etc. are faced with decision problems in transporting and storing of goods. Typical problems are transportation problems, network flow problems, traveling salesman problems, Chinese postman problems, vehicle routing planning problems and location problems (Rushton, Croucher, & Baker, 2017, p. 3ff). The students need to understand the problem itself and to learn how to formulate the problem mathematically, to solve it and to interpret the solution. A problem-based learning approach seems to be a suitable learning and teaching method for such tasks. Using real problems or realistic case studies, students should improve their problem-solving skills, their ability to independently acquire theoretical knowledge and develop critical thinking (Guimarães & Lima, 2021). In this context, suitable software is needed to solve the discussed problems.

Simulation software is an approach often used in Operations Management and Logistics and Supply Chain Management (LSCM) courses. A prominent example is the beer distribution game originally invented by Forrester in the 1960s (Dizikes, 2015) for which several software packages exist (Beergame.org, 2021). It deals with

the dynamics within a supply chain with one homogeneous good including the bullwhip effect (Edali & Yasarcan, 2016). Other examples of teaching and learning approaches using simulation software for LSCM are Ştefan, Hauge, Hasse, & Ştefan (2019), Kanet & Stößlein (2008), Angolia & Pagliari (2018) and the Fresh connection simulation (Inchainge, 2021). In most of these simulations, students have to make and enter various decisions within the supply chain, although there seems to be no direct software support for this. Therefore, simulation approaches for these logistical decisions often also require an optimisation software package.

A frequently used approach to solving logistical decisions are spreadsheet programs with add-ins for optimisation (Winkels, 2012). However, this approach is more suitable for small problems due to the limitation of the invoked solvers and its size restrictions. Furthermore, the students must have a high level of knowledge in operations research techniques to formulate, solve and interpret the problems where such programs do not allow an interactive and intuitive way to do it (Mason, 2013).

Another option are optimisation environments, especially mathematical programming languages such as the commercial AMPL (AMPL, 2021), GAMS (GAMS, 2021), etc. or open-source packages like GMPL (GMPL, 2021), CMPL (CMPL, 2021) or Pyomo (Pyomo, 2021). There are also packages that implement routines for selected logistical decisions with mathematical languages. For example, Matlog implements routines for transport problems, vehicle routing and network optimisation in Matlab (Kay, 2016). Furthermore, the Google OR-Tools are very interesting, which offer application programming interfaces (APIs) for various programming languages for various optimisation problems including selected logistical decision problems (Google, 2021b). As with spreadsheet solver add-ins, students need a high level of knowledge in operations research, which is not always the case, especially in management courses. (Grasas & Ramalhinho, 2016, p. 378).

Another possibility is the use of commercial logistics software in university courses. It is obvious that the use of such software gives students a realistic insight into the management of supply chains. In addition, the acquired skills have a positive impact on their future position in the labour market (Campbell, Goentzel, & Savelsbergh, 2000). However, commercial logistics software is often too expensive to purchase and maintain (Sweeney, Campbell, & Mundy, 2010, p. 298f). Some manufacturers offer cheaper or free versions of their software for academic purposes, although these are often limited in their functionalities or the size of the problems. For example,

anyLogistix offers a commercial „supply chain analytics software to design, optimize and analyze a companies' supply chain“ (anyLogistix, 2021a), for which a free version is available for academic purposes that does not include some of the commercial functionalities and is also limited in the number of customers and distribution centres and factories (anyLogistix, 2021b). If these limitations are not too restrictive for a particular course, then the use of free versions of commercial software makes sense. Another problem with commercial software packages can be that they do not offer the full range of decision problems discussed in university courses. A good example is Graphhopper (Graphhopper, 2021), which offers an excellent API for vehicle routing. However, it only supports vehicle routing. If other problems are to be discussed, additional software is needed.

Academic logistical software, which offers a wide range on supported logistical decision problems, is rather rare. As an example, the Toolbox of the Technical University of Dresden can be mentioned (Toolbox, 2013). In addition to problems in procurement logistics and production logistics, shortest-path problems, network flow problems, transport problems, vehicle routing problems and location problems can also be solved. However, the software is only available in German, seems rather orientated on techniques and does not support all relevant aspects of logistical decision making. Another example is a web-based decision support system for vehicle routing problems developed by Gr̄asas & Ramalhinho (Gr̄asas & Ramalhinho, 2016). Embedded in a problem-based learning approach, the tool is intended “to better understand the solution method and to show students the need for decision-making software in complex problems” (Gr̄asas & Ramalhinho, 2016, p. 389). The combination of the easy-to-use web-based tool and the didactic approach is very interesting. However, this tool can only be used for vehicle routing problems.

The motivation for the development of LogisticsLab was, considering all the possibilities of using software for teaching logistical decision problems, to create an academic software that is easy and intuitive to use and offers a wide range of supported logistical decision problems.

This paper describes LogisticsLab. In the first section, an overview of the software including the main functionalities and the file formats are given. The following section is intended to illustrate how students can solve realistic case studies by using LogisticsLab. The described case studies have been used in several courses at Bachelor level in German and international universities. The paper ends with a summary.

LOGISTICS LAB

Overview

LogisticsLab is an academic software for decision support in logistics. This software can be classified as a problem-oriented decision support system that can be used to model and solve transportation problems, network flow

problems, traveling salesman problems, Chinese postman problems, vehicle routing planning problems and location problems in different variations.

LogisticsLab was originally developed in the 1990s under the name *EUS-Lehrsoftware* by Dieter Feige for logistics courses at the Friedrich-Alexander University of Erlangen-Nuremberg and fundamentally revised and expanded by the author of this article.

LogisticsLab is provided under the LogisticsLab Academic License. It is available for academic use (such as research and teaching or for reproducing case studies in academic textbooks), free of charge and without warranty. It can be downloaded at <http://logisticslab.org> and is available for Microsoft Windows. No installation is required. The user only has to unzip the downloaded zip file and the contained executables can be used immediately.

Main functionalities

After unpacking the downloaded distribution file, the LogisticsLab folder contains several binary files with specific names that provide the following functionalities.

TPP

LogisticsLab/TPP is a software that can be used to solve classical transport problem (Hillier & Lieberman, 2015, p. 319ff) and bottleneck transport problems (Garfinkel & Rao, 1971). Minimisation and maximisation (or MiniMax and MaxiMin for bottleneck problems) can be selected for the objective function sense. It is also possible to invoke step-fixed costs for source-destination-relations. In addition, various constraints can be taken into account (capacities, single-sourcing, supply or demand surplus, etc.).

NWF

LogisticsLab/NWF is intended for solving network flow problems, which can be modelled as min-cost flow problems (Ghiani, Laporte, & Musmanno, 2013). One of the most important problems to be solved with this model is the transshipment problem (Hillier & Lieberman, 2015, p. 401).

TSP

LogisticsLab/TSP is used for solving traveling salesman problems (Applegate, Bixby, Chvátal, & Cook, 2006) and open traveling salesman problems (Steglich, Feige, & Klaus, 2016, p. 312ff) for symmetrical and asymmetrical distances between the nodes to be approached. For open traveling salesman problems, four different types can be specified (fixed start and end node, fixed start node and free end node, fixed end node and free start node, free start and end node). Symmetric distances can be calculated as airline distances using the coordinates of the nodes as Euclidian distances, Manhattan distances and great circle distances. It is also possible to define directed or undirected edges with given distances. In addition, a distance matrix containing distances obtained by APIs of geographic information systems like Google Maps (Google, 2021a), Bing Maps (Microsoft, 2021) or

OpenStreetMap (OpenStreetMap, 2021b) can be imported via LogisticsLab's file format.

CPP

LogisticsLab/ CPP is intended for solving Chinese postman problems in undirected, directed and mixed networks. Starting from a start node, each edge is to be traversed at least once and returned to the start node at the end of the round trip. The edge-oriented round trip to be found should minimise the total distance (or other adequate evaluation) (Steglich, Feige, & Klaus, 2016, p. 320ff).

VRP

LogisticsLab/VRP is a software with which vehicle routing problems in different variants can be solved. The capacitated vehicle routing problem (VRP) aims to find a set of routes at a minimal cost level, beginning and ending the routes at a depot, so that the demands of all nodes (customers, cities, etc.) are satisfied. Each node is visited at least once and each vehicle has a limited capacity (Williams, 2013, S. 198ff). With VRP, a user is able to solve such problems in symmetric graphs using airline distances (Euclidian, Manhattan and great circle distances), distances defined for edges or distances obtained by geographic information systems (e.g. Google Maps, Bing Maps, OpenStreetMap). Several additional constraints can be invoked into the decision (route duration, route lengths, etc.)

WLP

Warehouse Location Problems are discrete location problems where both fixed location costs and variable transport costs as well as the capacities of the potential sources and the demands of the destinations are included in the location decision. The objective is to minimise the sum of the transportation costs and the fixed costs of building and running warehouses by deciding which warehouses are established and which customer is delivered to by which warehouse (Fernández & Landete, 2015, S. 47ff). LogisticsLab/WLP is designed as a software for solving single-stage capacitated and uncapacitated Warehouse Location Problems, always assuming that a demand node can only be supplied from one location at a time (single-sourcing).

CLP

LogisticsLab/CLP is a software to solve continuous (and discrete) median and centre problems. Continuous location problems determine the best possible greenfield location. The point where a planned logistics node can be optimally placed is to be found within a certain area on a flat or spherical surface. In contrast, in discrete location problems, only existing logistics nodes can act as locations (Steglich, Feige, & Klaus, 2016, p. 394ff). The following two approaches are distinguished: The *median problem* aims at minimising the sum of the (weighted) distances between the sources and the destinations and consists in finding the optimal locations of the sources and an optimal assignment of the destinations to the

sources. Depending on the number p of locations sought, these problems are also called p -median problems (Neema, Maniruzzaman, & Ohgai, 2011). The *centre problem* aims at minimising the maximum (weighted) distance over all relations between the locations of the sources and the assigned destinations. Depending on the number p of locations to be determined, these problems are also called p -centre problems (Calik, Labbé, & Yaman, 2015).

User Interface

The user interface of each part of LogisticsLab follows a uniform structure, as shown in Figure 1 for LogisticsLab/VRP as an example. In addition to the usual elements, the user interface has a network area on the left-hand side of the application window with the graphical representation of the network and a data area on the right-hand side in which several areas can be selected via tabs and filled with the necessary data. Generally, interactive data entry and modification takes place in the data area, but it is also possible to modify some of the data in the network area interactively. The network and the data area can be spread over the entire window.

File Formats

The data of the programs are stored in tab separated text files with a specific format. To distinguish the data for the different programs, the files are given the program abbreviation as file extensions, i.e.: TPP, NWF, TSP, CPP, VRP, WLP and CLP.

The following principles apply to data files:

- Each file begins with the keyword for the program for identification. This keyword can be followed in the same line by information on the date and time.
- The next line contains a comment.
- For some of the applications, a section for options follows. This begins with the keyword OPTIONS, after which the corresponding options are listed in the following lines.
- The problem data is divided into data sections. Each section begins with a line in which the corresponding keyword (e.g., SOURCES, DESTINATIONS) for the respective data must appear. In most cases, the size specifications for the data section (e.g., number of sources and destinations) follow in a subsequent line. The following lines contain the corresponding data. The records are defined line-wise as ASCII text. The data entries are separated with tabulators.
- The end of file is expressed with the keyword EEE in the last line.

The data files can be created via the menu function *Save Problem* or *Save Problem as*. However, it is also possible to edit and maintain an existing data file (as well solution files) with a spreadsheet software (e.g., Microsoft Excel or LibreOffice/Calc). Finally, the data must be saved as a pure text file with data separation by tabs.

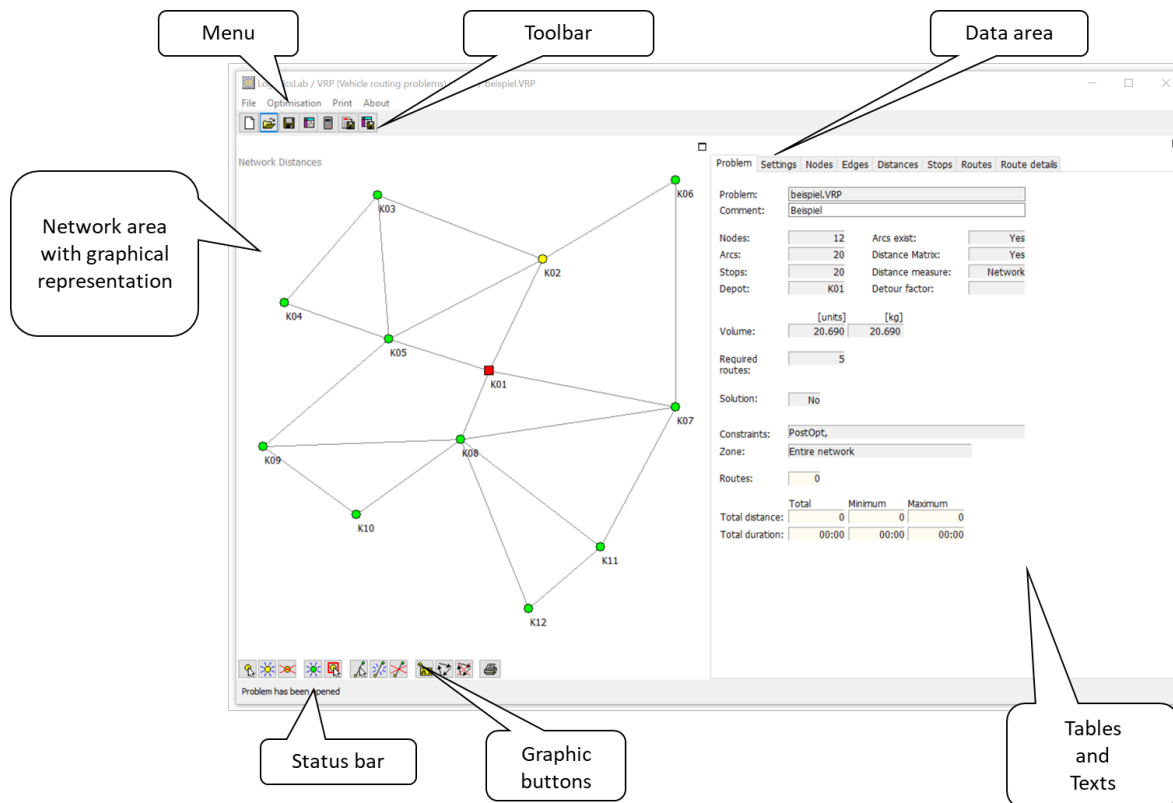


Figure 1: User interface of LogisticsLab/VRP

SELECTED CASE STUDIES

This section is intended to illustrate how students can solve realistic case studies by using LogisticsLab. Both case studies have been used in courses at Bachelor level at several European universities, often using a problem-based learning and teaching approach (Hung, 2011) (Grasas & Ramalhinho, 2016).

In a first step, the students, organised in groups, have to analyse a given realistic problem and work out the theoretical basis of the problem in order to formulate a suitable mathematical model. The mathematical model helps the students to understand and formulate the objective and the constraints of the problem and what data is needed to solve the case. In addition, students have to obtain the problem data from various sources. Afterwards, the problem, including the obtained data, has to be entered into LogisticsLab and solved with it. The last but not least task is the interpretation and presentation of the solution found.

Street condition survey in Manhattan

In this case study, the students were asked to find the shortest route to examine the condition of part of the streets in Manhattan, NY. As shown in Figure 2, it is the area bounded on the northwest by the 10th Avenue/West 57th Street intersection and on the southeast by the 6th Avenue/East 42nd Street intersection. All streets in this area are to be driven at least once.

This problem can be solved as a Chinese postman problem which can be understood as an edge-oriented round-trip problem. Consider a network $G = (N, E)$, where N is a set of nodes and E is a set of edges joining pairs of nodes. Each edge has to be used at least once, whereby the tour starts and ends at a specific node. The total distance of this route is to be minimised (Ahuja, Magnanti, & Orlin, 2013, S. 740) (Steglich, Feige, & Klaus, 2016).

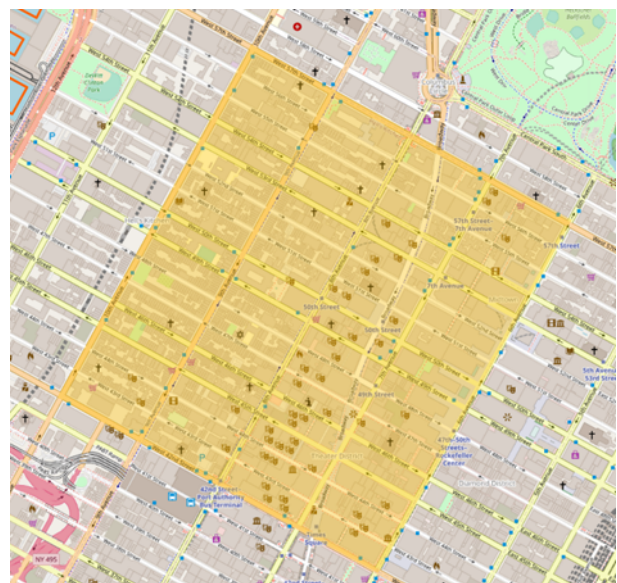


Figure 2: Area in Manhattan (OpenstreetMap, 2021a)

In this case, the nodes are defined by the street intersections and the edges by the streets, considering the directions of the streets. The coordinates of the nodes and the distances of the edges can be obtained by using the APIs of geographical information systems like Google Maps, Bing Maps or OpenStreetMap.

After starting the application, the user interface shown in Figure 3 appears, which, like all other components of LogisticsLab, consists of a network area and a data area.

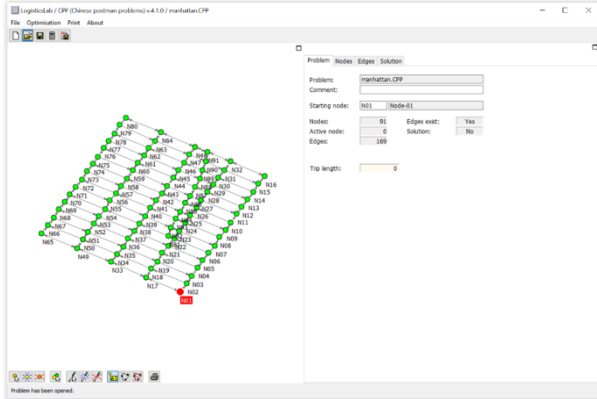


Figure 3: User interface of LogisticsLab/CPP

The left side of the CPP window is used to visualise the network graph depending on the data entered. If the network graph contains connections between the nodes, these are shown as edges. Directed edges are represented by arrows. Undirected edges do not have arrowheads. The data area on the right-hand side of the CPP user interface contains four worksheets for entering data and outputting optimisation results (*Problem*, *Nodes*, *Edges*, *Solution*).

The first step using LogisticsLab/CPP is to generate a new problem. To do this, the menu item *File* → *New Problem* or the *New Problem* button in the toolbar is selected. In addition to a comment, the number of *nodes* (in this case 91 nodes) and the maximum distance between the nodes (*Max. distance*) must be entered (Figure 4). The latter is used when generating randomly based coordinates (*Coordinates* → *Randomly*). The generated coordinates can be edited subsequently.

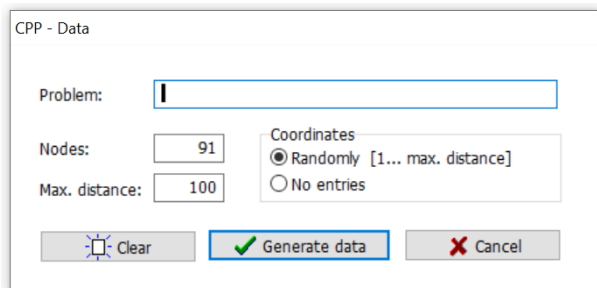


Figure 4: Creating a problem in LogisticsLab/CPP

After generating the problem, the data can be entered in the data area. The nodes can be edited in the *Nodes* tab (Figure 5).

A user can either enter the number of nodes or add each node individually to the list with the *Add* button.

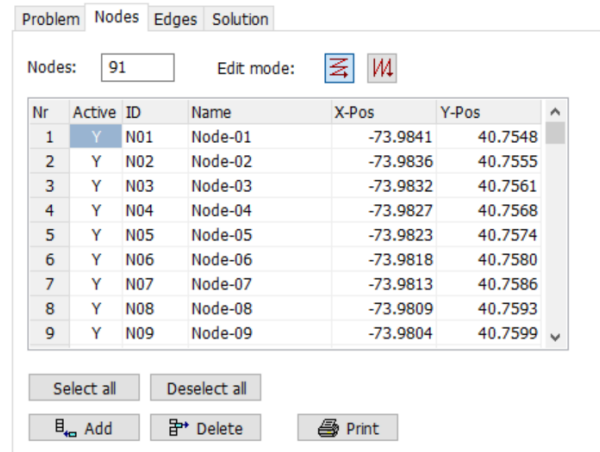


Figure 5: *Nodes* tab in LogisticsLab/CPP

Each node is specified by a selection (*Active* - selection for a calculation), a unique node ID (*ID*), a node name (*Name*) and an X and a Y coordinate. Geographical coordinates can also be used instead of simple X and Y coordinates. In this case, the longitude must be entered as the X-value and the latitude as the Y-value.

An easier way is to open the generated CPP file in a spreadsheet software, edit the names and copy and paste the coordinates found with Google Maps, Bing Maps or OpenStreetMap into the file (Figure 6).

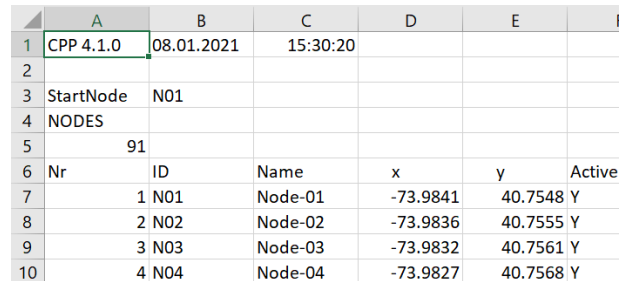


Figure 6: Editing a CPP file in Microsoft Excel

The edges have to be defined in the *Edges* tab (Figure 7). To do this, the node from which the edges start must be specified in the *From node* field. The edges are defined in the *To nodes* list by entering the name, length and type of the edge. The type can be *U* for an undirected edge or *D* for a directed edge.

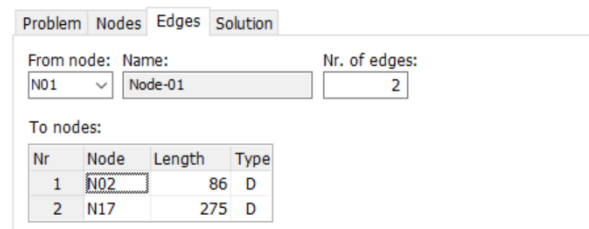


Figure 7: *Edges* tab in LogisticsLab/CPP

For large problems, it is also recommended for the edges to open the CPP file in a spreadsheet program and copy and paste the distances into it.

After entering all the data, the problem can be solved by selecting either the *Optimisation* → *Start Optimisation* menu or the *Optimise* button in the toolbar. To solve a Chinese postman problem, the following two steps are carried out by LogisticsLab/ CPP automatically (Steglich, Feige, & Klaus, 2016, p. 320 ff):

1. Cost- or distance-minimal extension of the network into a Euler network,
2. Determination of the Euler tour.

The cost- or distance-minimal extension of the existing network is solved on the basis of an integer linear optimisation model (Winkels, 2012, p. 590f) (Steglich, Feige, & Klaus, 2016, p. 325f), the results of which are used by a recursive algorithm to determine an Euler tour (Mattfeld & Vahrenkamp, 2014, p. 223f) (Steglich, Feige, & Klaus, 2016, p. 322ff) and thus to solve the Chinese postman problem.

After solving the problem, the graphical representation of the solution appears in the *Network* area (Figure 9) and the details in the *Solution* tab (Figure 8). In the Network area (Figure 9), the thickness of the edges indicates whether they have to be traversed several times.

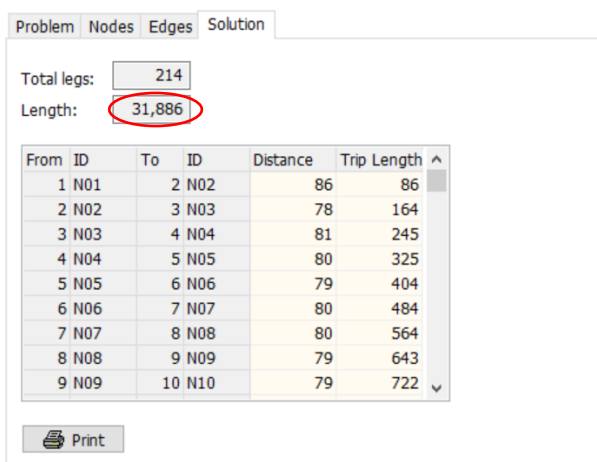


Figure 8: *Solution* tab in LogisticsLab/ CPP

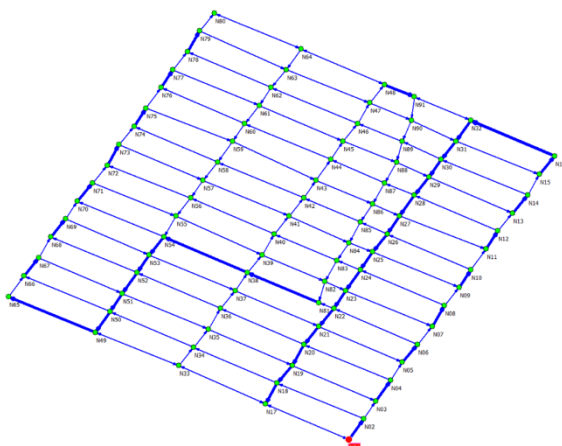


Figure 9: *Network area* in LogisticsLab/ CPP

The best solution found has a length of 31.886 km (Figure 8). Both the graphical solution and the list of the entire route can be printed. It is also possible to save the solution in a solution file which can easily be imported into a spreadsheet program.

Planning three disaster relief centres in the state of Brandenburg, Germany

In this case study, students were asked to find the optimal locations of three disaster relief centres for the state of Brandenburg. The task of the centres is to send technology and emergency forces to the towns and municipalities as fast as possible in the event of a disaster. The maximum distance of the centres to the potential places of operation is to be minimised. Since cities and municipalities with many inhabitants are to be given priority in the event of a disaster, the distances are to be weighted according to the number of inhabitants. The coordinates and the number of inhabitants of the cities and municipalities can be taken from data of the German Federal Statistical Office (Statistisches Bundesamt, 2020). The locations of the disaster relief centres have to be planned on greenfield sites. The problem is a continuous p -centre problem without considering capacities and with demand-weighted distances, which can be formulated as follows (Drezner, 1984) (Drezner, 2011):

$$R \rightarrow \min! \quad (1)$$

s. t.

$$b_j \cdot x_{ij} \cdot d((u_i, v_i), (x_j, y_j)) \leq R ; i \in S, j \in D \quad (2)$$

$$\sum_{i \in S} x_{ij} = 1; j \in D \quad (3)$$

$$x_{ij} \in \{0,1\} ; i \in S, j \in D \quad (4)$$

Indices and sets

$i \in S$ sources

$j \in D$ destinations

Parameters

b_j demand of destination j

x_j, y_j coordinates of destination j

Function

$d(\cdot)$ distance function

Variables

R maximum demand weighted distance

x_{ij} assignment variable whether source i serves destination j

u_j, v_j coordinates of source i

According to the expressions (1) and (2), the maximum demand-weighted distance R between the sources and the destinations is to be minimised. These distances are to be determined for all combinations of sources and destinations with a suitable distance function $d(\cdot)$. These distances are only relevant if the corresponding assignment variables x_{ij} are equal to the value one, i.e., destination j is assigned to source i . Expression (3) ensures that a destination is assigned to exactly one source (Steglich, Feige, & Klaus, 2016, p. 411).

The problem can be solved with LogisticsLab/CLP, which, like all other parts of LogisticsLab, has a user interface consisting of a network area and a data area (Figure 10).

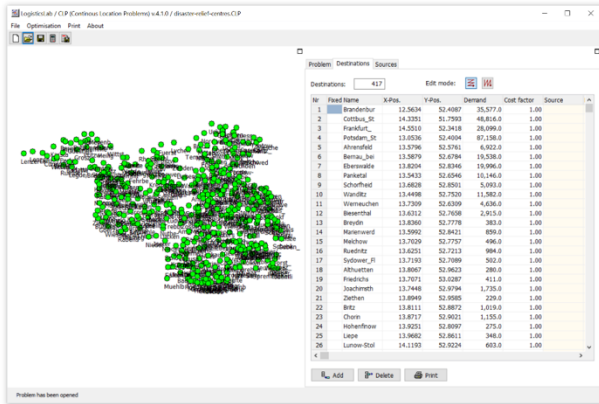


Figure 10: User interface of LogisticsLab/CLP

The first step is to create the problem file and to collect the problem data. Since the state of Brandenburg consists of 417 towns and municipalities, a problem with 417 destinations is to be created via the menu item *File* → *New Problem* or the *New Problem* button in the toolbar, as shown in Figure 11.

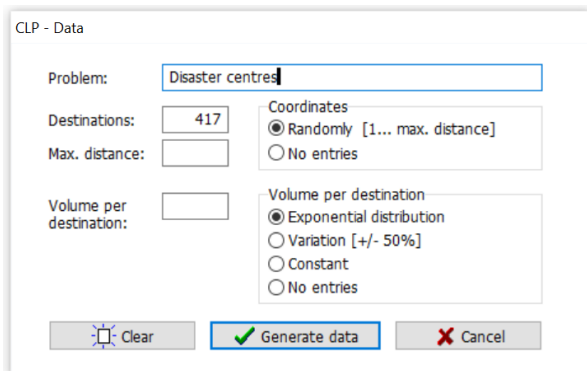


Figure 11: Creating a problem in LogisticsLab/CLP

Afterwards, the data of the generated problem can be entered in the data area (Figure 12). For all destinations, the names, the coordinates and the demands have to be entered, whereby for this problem the demands are defined by the number of inhabitants of the corresponding town or municipality. Since geographical coordinates are used, the longitude must be entered as the X-value and the latitude as the Y-value. The names, the coordinates and the inhabitants of the towns and municipalities can be taken from an Excel file of the German Federal Statistical Office (Statistisches Bundesamt, 2020). However, it is recommended to copy the mentioned data from this Excel file and paste it into the CLP file opened in a spreadsheet program (Figure 13).

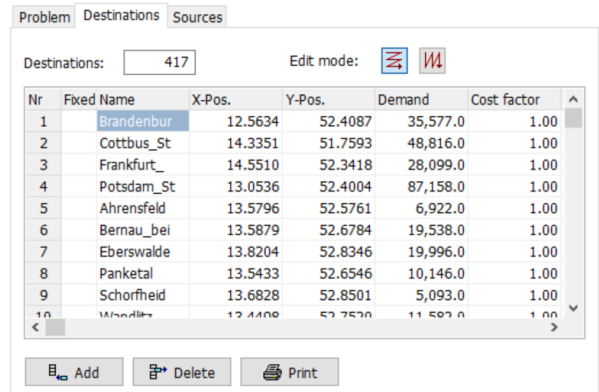


Figure 12: Destination tab in LogisticsLab/CLP

	A	B	C	D	E	F
1	CLP 4.0.5	02.02.2021	10:14:46			
2	Disaster centres					
3	OPTIONS					
4	Distance	GreatCircle				
5	DESTINATIONS					
6		417				
7	Name	Fixed	x	y	Demand	CostFactor Sour
8	Brandenburg N		12.563364	52.408651	35 577	1
9	Cottbus_Stac N		14.335117	51.759345	48 816	1
10	Frankfurt_(ON		14.550979	52.341757	28 099	1
11	Potsdam_Sta N		13.05358	52.400355	87 158	1
12	Ahrensfelde N		13.57961	52.576115	6 922	1
13	Bernau_bei_N		13.587912	52.678368	19 538	1
14	Eberswalde_N		13.820369	52.834577	19 996	1

Figure 13: CLP problem file in Excel

The *Sources* tab (Figure 14) contains the details of the locations whose positions are to be determined by the optimisation. By entering the number of sources, the size of the input sheet is adjusted automatically. The names and the capacities (if needed) can be edited in this tab.

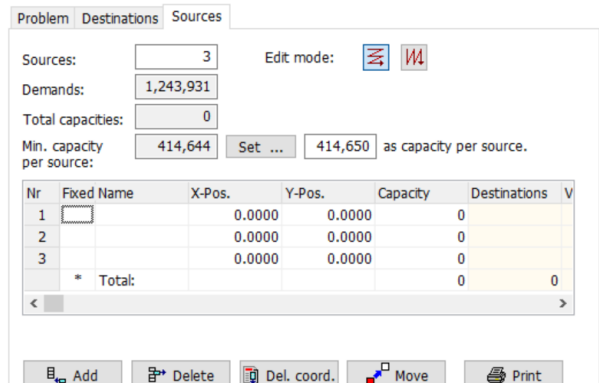


Figure 14: Sources tab in LogisticsLab/CLP

Once all the data has been entered, the problem can be solved. To do this, the problem type *centre of gravity* and the distance function to be used (*Distance* → *Great Circle*) must be selected in the *Problem* tab (Figure 15). Since the number of inhabitants of the destinations influence the priority in the event of a disaster, it must be specified that the distances are weighted with the demands of the destinations (checkbox *Demand weighted*).

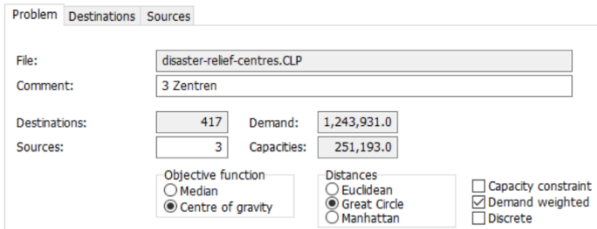


Figure 15: Problem tab in LogisticsLab/CLP

Optimisation is started by selecting either the *Optimisation* → *Start Optimisation* menu or the *Optimise* button in the toolbar. LogisticsLab uses state of the art heuristics to solve continuous location problems. As is usual for heuristics, the solutions may vary from run to run.

The solution of the problem is displayed in both the network area and data area. In the network area (Figure 16), the locations found for the three disaster relief centres and additionally the assignments of the towns and municipalities as potential places of a disaster event are shown. The exact locations of the disaster relief centres can be found in the *Sources* tab (Figure 17) in the columns *X-Pos* and *Y-Pos*. Besides to the number of assigned destinations (*Destinations*), the nearest destinations to the greenfield locations of the sources are shown.

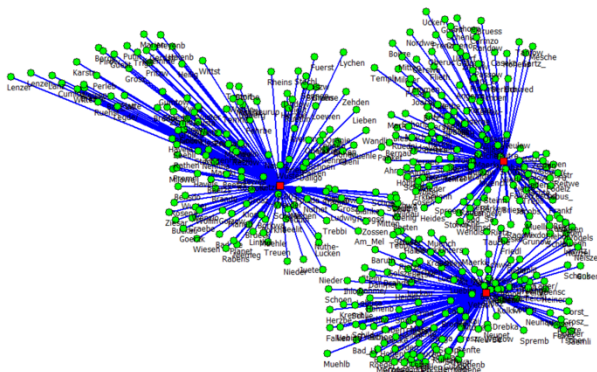


Figure 16: Network area in LogisticsLab/CLP

Nr	Fixed Name	X-Pos.	Y-Pos.	Capacity	Destinations	Volume	Costs	Nearest dest.	Distance
1	S.001	14.2625	52.5697	83.731	131	320,295.0	13,188,856.1	Neuhardenb.	3.2
2	S.002	14.1566	51.7918	83.731	129	264,879.0	8,615,015.3	Werbent/We	3.0
3	S.003	12.9436	52.4251	83.731	157	658,757.0	24,043,457.6	Weeder_Hb	5.5
* Total:				251,193	417	1,243,931.0	45,847,329.0		

Figure 17: Solution in the Sources tab

The objective was to minimise the maximum of the demand-weighted distances between the locations to be found and the destinations. As shown in Figure 18, the objective function value equals 922,193.6 person-kilometres (similar to passenger-distance). This value is the minimum *Maximum Costs* of all relations between the new locations of the disaster relief centres and the assigned destinations. It results from the multiplication of the number of the inhabitants (*Demand*) and the distances. The cost factor is set to the value one and can be

ignored. This indicator is reasonable because it is assumed that an increasing number of inhabitants also leads to the increase of the amount of technology and emergency forces to be sent to the location of a disaster event.

Initial solution	Solution Nr.	Objective value	Cal. time
		1,856,465.40	
First solution:			
Current solution:	150	1,788	1,610,072.70
Best solution:	144	1,708	922,193.60

Transport costs:	45,847,329.00	[units*cost factor*km]	Capacity overflow:
Transport distance:	18,613.40	[km]	
Transport volume:	1,243,931.00	[units]	

Distances:	Average	Minimum	Maximum	[km]
	44.6	3.0	128.5	
Costs:	109,945.6	1,493.4	922,193.6	[units*cost factor*km]
Destinations per source:	139.0	129.0	157.0	
Volume per source:	414,643.7	264,879.0	658,757.0	[units]

Figure 18: Key indicators in the Problem tab

The assignments of the destinations to the newly found locations of the sources can be found in the *Destinations* tab (Figure 19). The relation with the maximum inhabitant weighted distances is between the destination *Bernau* (Nr. 6) and the first source which is located nearby *Neuhardenberg* (Figure 19 and Figure 17).

Nr	Fixed Name	X-Pos.	Y-Pos.	Demand	Cost factor	Source	Distance	Costs
1	Brandenbur	12.5634	52.4087	35,577.0	1.00	S.003	25.9	921,444.3
2	Cottbus_St	14.2351	51.7593	48,816.0	1.00	S.002	12.8	624,844.8
3	Frankfurt_	14.5510	52.3418	28,099.0	1.00	S.001	32.0	899,168.0
4	Potsdam_St	13.0536	52.4004	87,158.0	1.00	S.003	8.0	697,264.0
5	Ahrensfeld	13.5796	52.5761	6,922.0	1.00	S.001	46.2	319,796.4
6	Bernau_bei	13.5879	52.6784	19,538.0	1.00	S.001	47.2	922,193.6
7	Eberswalde	13.8204	52.8346	19,996.0	1.00	S.001	41.9	837,832.4
8	Panketal	13.5433	52.6546	10,146.0	1.00	S.003	48.0	487,008.0
9	Schorfheid	13.6828	52.8501	5,093.0	1.00	S.001	50.0	254,650.0
10	Wandlitz	13.4498	52.7520	11,582.0	1.00	S.003	50.0	579,100.0
11	Werneuchen	13.7309	52.6309	4,636.0	1.00	S.001	36.6	169,677.6

Figure 19: Solution in the Destinations tab

The objective of this case study can be changed if it is assumed that the amount of technology and emergency forces is the same for all events and therefore independent of the number of inhabitants. In this case, only the maximum over all distances between the locations and the potential disaster event places is to be minimised. This can be achieved simply by not activating the option *Demand weighted* in the *Problem* tab (Figure 15).

After running the optimisation again, a new solution can be found where the locations of the three disaster relief centres are not attracted to destinations with high populations. This can be seen in the graphical solution in the network area (Figure 20) and on the newly determined coordinates of the sources in the *Sources* tab (Figure 21).

Since a heuristic is used, the solutions may be different for different optimisation runs. The minimal maximum distance found equals 73.2 kilometres (Figure 22). That means that the teams have to travel a maximum of 73.2 kilometres over all relations between the newly determined locations of the disaster relief centres and the potential places of a disaster event. This is less than the minimal maximum distance of the original problem which is equal to 128.5 kilometres (Figure 18 → *Distances* → *Maximum*).

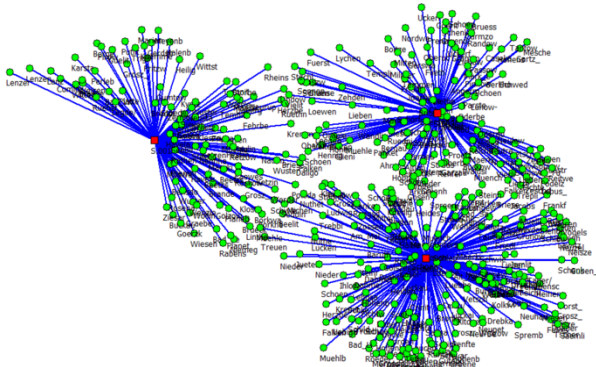


Figure 20: Graphical solution of the unweighted case in the network area in LogisticsLab/CLP

Nr	Fixed Name	X-Pos.	Y-Pos.	Capacity	Destinations	Volume	Costs	Nearest dest.	Distance
1	S.001	12.2190	52.6636	83,731	106	344,148.0	4,785.2	Havelaue	7.6
2	S.002	13.8301	51.9628	83,731	182	523,642.0	8,006.0	Schoenwald	4.8
3	S.003	13.8991	52.8236	83,731	129	376,141.0	4,905.9	Hohenfinow	2.3
* Total:				251,193	417	1,243,931.0	17,697.1		

Figure 21: New locations of the unweighted case in the Sources tab in LogisticsLab/CLP

	Initial solution	Solution Nr.	Objective value	Cal. time	
First solution:			138.60		
Current solution:	150	7,668	91.00	16.09 [sec]	Improvements
Best solution:	5	539	73.20	1.30 [sec]	47.19 [%]

Transport costs:	17,697.10	[cost factor*km]	Capacity overflow:	
Transport distance:	17,697.10	[km]		
Transport volume:	1,243,931.00	[units]		

	Average	Minimum	Maximum	
Distances:	42.4	2.3	73.2	[km]
Costs:	42.4	2.3	73.2	cost factor*km
Destinations per source:	139.0	106.0	182.0	
Volume per source:	414,643.7	344,148.0	523,642.0	[units]

Figure 22: Key indicators in the Problem tab for the unweighted case

However, the locations found with a continuous p -centre problem cannot be implemented in reality in every case. So, it could be possible that the three locations nearby *Havelaue*, *Schoenwalde* and *Hohenfinow* (Figure 21) cannot be realised for many reasons. If only existing nodes in a network are available as locations for the sources, a discrete p -centre problem has to be solved.

In difference to a continuous problem, the coordinates of the locations are not variables, because only predefined nodes of the set S of potential sources can be used as locations. Additionally, all distances between the potential sources $i \in S$ and the destinations $j \in D$ are known. Regarding the mathematical model, please refer to the literature (Calik, Labbé, & Yaman, 2015, p. 83f).

LogisticsLab/CLP is mainly intended to solve continuous p -median and p -centre problems. However, it offers also to solve both as discrete location problems using a heuristic. To do so, a user only has to choose the option *Discrete* in the *Problem* tab (Figure 15) and to start the optimisation again. After selecting this option, a new solution can be found for the given unweighted case. It

should also be mentioned for the discrete problem that the solutions of different optimisation runs may differ due to the non-deterministic nature of the underlying heuristic. The discrete solution is shown as graphical solution (Figure 23) in the network area and the coordinates are shown in the *Sources* tab (Figure 24).

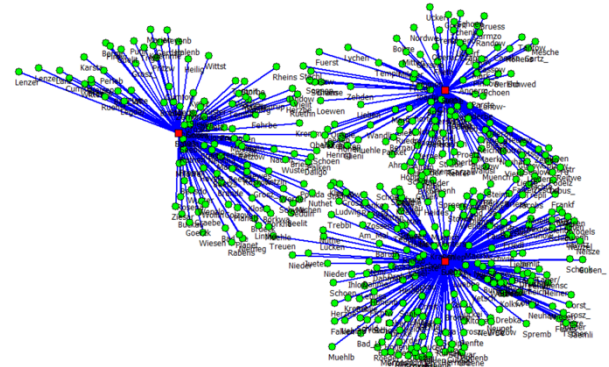


Figure 23: Graphical solution of the discrete case in the network area in LogisticsLab/CLP

It is interesting to note that the new locations for the disaster relief centres in *Ziethen*, *Luebben* and *Havelaue* are partially unequal to *Havelaue*, *Schoenwalde* and *Hohenfinow*, which are the closest destinations to the locations found for the continuous problem. Moving the continuous locations to the closest destinations may lead to suboptimal solutions.

Nr	Fixed Name	X-Pos.	Y-Pos.	Capacity	Destinations	Volume	Costs	Nearest dest.	Distance
1	S.001	13.8949	52.9585	83,731	118	349,866.0	4,519.3	Ziethen	0.0
2	S.002	13.8966	51.9420	83,731	186	521,105.0	8,125.7	Luebben_(S	0.0
3	S.003	12.3106	52.7037	83,731	113	372,960.0	4,947.1	Havelaue	0.0
* Total:				251,193	417	1,243,931.0	17,592.1		

Figure 24: New locations of the discrete case in the Sources tab in LogisticsLab/CLP

The minimal *Maximum Distance* found for the discrete problems is equal to 75.6 kilometres (Figure 25). This is only 2.4 kilometres more for this more realistic solution compared to the solution of the continuous problem.

	Initial solution	Solution Nr.	Objective value	Cal. time	
First solution:			111.30		
Current solution:	150	6,855	89.80	42.21 [sec]	Improvements
Best solution:	2	78	75.60	0.50 [sec]	32.08 [%]

Transport costs:	17,592.10	[cost factor*km]	Capacity overflow:	
Transport distance:	17,592.10	[km]		
Transport volume:	1,243,931.00	[units]		

	Average	Minimum	Maximum	
Distances:	42.2	0.0	75.6	[km]
Costs:	42.2	0.0	75.6	cost factor*km
Destinations per source:	139.0	113.0	186.0	
Volume per source:	414,643.7	349,866.0	521,105.0	[units]

Figure 25: Key indicators in the Problem tab for the unweighted case

SUMMARY

Students who are confronted with logistical decision problems in several courses need appropriate software to solve and interpret realistic case studies. The frequently used solver add-ins in spreadsheet programs or general optimisation software require a high level of knowledge in operations research and do not always allow intuitive access. Commercial logistics software and the rather rare academic software often offer only a smaller selection of supported logistical decisions.

This article described the academic software LogisticsLab, which supports a wide range of logistical decisions. The idea of the software is the systematic combination the experiences and intuitions of human decision-makers with the opportunities of computer-assisted modelling and optimisation in the form of an interactive decision support system. Two realistic case studies, used in courses at several European universities, demonstrate how LogisticsLab can be used to gain a better understanding of logistical decision-making in an interactive and intuitive way.

ACKNOWLEDGEMENTS

The author would like to thank Dieter Feige for the many years of cooperation and for the opportunity to continue the LogisticsLab project. Unfortunately, Dieter Feige passed away in 2019, so the author has lost not only a colleague but also a friend. This article is dedicated to him.

REFERENCES

- Ştefan, I. A., Hauge, J. B., Hasse, F., & Ştefan, A. (2019). Using Serious Games and Simulations for Teaching Co-Operative Decision-making. *7th International Conference on Information Technology and Quantitative Management (ITQM 2019)* (pp. 745-753). Procedia Computer Science.
- Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (2013). *Network Flows: Theory, Algorithms, and Applications*. Harlow: Pearson New International Edition.
- AMPL (2021). Retrieved February 2021, <https://ampl.com>.
- Angolia, M. G., & Pagliari, L. R. (2018). Experiential Learning for Logistics and Supply Chain Management Using an SAP ERP Software Simulation. *Decision Sciences Journal of Innovative Education*, 16(2), 104-125.
- anyLogistix (2021a). *anyLogistix - Features overview*. Retrieved April 2021, <https://www.anylogistix.com/alx-features/>.
- anyLogistix (2021b). *anyLogistix - Personal Learning Edition*. Retrieved April 2021, <https://www.anylogistix.com/personal-learning-edition/>.
- Applegate, D., Bixby, R., Chvátal, V., & Cook, W. (2006). *The traveling salesman problem: A Computational Study*. Princeton: Princeton University Press.
- Beergame.org (2021). *Versions*. Retrieved April 2021, <https://beergame.org/the-game/versions/>.
- Calik, H., Labbé, M., & Yaman, H. (2015). p-Center Problems. In G. Laporte, S. Nickel, & F. Saldanha da Gama (Eds.), *Location Science* (pp. 79-92). Cham et al.: Springer.
- Campbell, A., Goentzel, J., & Savelsbergh, M. (2000). Experiences with the use of supply chain management software in education. *Production and Operations Management*, 9(1), 66-80.
- CMPL (2021). Retrieved February 2021, <http://coliop.org>.
- Dizikes, P. (2015). *The Many Careers of Jay Forrester*. Retrieved April 2021, <https://www.technologyreview.com/2015/06/23/167538/the-many-careers-of-jay-forrester/>.
- Drezner, Z. (1984). The p-Centre Problem-Heuristic and Optimal Algorithms. *The Journal of the Operational Research Society*, 35, 741-748.
- Drezner, Z. (2011). Continuous Center Problems. In H. Eiselt, & V. Marianov (Eds.), *Foundations of Location Analysis* (pp. 63-78). Springer US.
- Edali, M., & Yasarcan, H. (2016). Results of a beer game experiment: Should a manager always behave according to the book? *21(S1)*, 190-199.
- Fernández, E., & Landete, M. (2015). Fixed-Charge Facility Location Problems. In G. Laporte, S. Nickel, & F. Saldanha da Gama (Eds.), *Location Science* (pp. 47-78). Cham et al.: Springer.
- GAMS (2021). Retrieved February 2021, <https://www.gams.com>.
- Garfinkel, R., & Rao, M. (1971). The bottleneck transportation problem. *Naval Research Logistics Quarterly* (18), 465-472.
- Ghiani, G., Laporte, G., & Musmanno, R. (2013). *Introduction to Logistics Systems Management* (2. ed.). Chichester: Wiley.
- GMPL (2021). *GNU MathProg modeling language*. Retrieved February 2021, <https://www.gnu.org/software/glpk/>.
- Google (2021a). *Distance Matrix API*. Retrieved February 2021, <https://developers.google.com/maps/documentation/distance-matrix/overview>.
- Google (2021b). *Google OR-Tools*. Retrieved September 2021, <https://developers.google.com/optimization>.
- Graphhopper (2021). *Products*. Retrieved February 2021, <https://www.graphhopper.com/products/>.
- Grasas, A., & Ramalhinho, H. (2016). Teaching distribution planning: a problem-based learning approach. *The International Journal of Logistics Management*, 27(2), 377-394.
- Guimarães, L., & Lima, R. (2021). Changes in teaching and learning practice in an undergraduate logistics and transportation course using problem-based learning. *Journal of University Teaching & Learning Practice*, 18(3), 11-27.
- Hillier, F., & Lieberman, G. (2015). *Introduction to Operations Research* (10. ed.). New York et al.: McGraw-Hill.
- Hung, W. (2011). Theory to reality: a few issues in implementing problem-based learning. *Education Tech Research Dev*(59), 529-552.

- Inchainge (2021). *Discover The Fresh Connection*. Retrieved April 2021, <https://inchainge.com/business-games/tfc/>
- Kanet, J. J., & Stößlein, M. (2008). Using a Supply Chain Game to Effect Problem-Based Learning in an Undergraduate Operations Management Program. *Decision Sciences Journal of Innovative Education*, 6(2), 287-295.
- Kay, M. G. (2016). Matlog: Logistics Engineering using Matlab. *Journal of Engineering Sciences and Design*, 4(1), 15-20.
- Mason, A. J. (2013). SolverStudio: A New Tool for Better Optimisation and Simulation Modelling in Excel. *INFORMS Transactions on Education*, 45-52.
- Mattfeld, D., & Vahrenkamp, R. (2014). *Logistiknetzwerke: Modelle für Standortwahl und Tourenplanung* (2. ed.). Wiesbaden: Springer Gabler.
- Microsoft (2021). *Distance Matrix API*. Retrieved February 2021, <https://www.microsoft.com/en-us/maps/distance-matrix>.
- Neema, M., Maniruzzaman, K., & Ohgai, A. (2011). New Genetic Algorithms Based Approaches to Continuous p-Median Problem. *Networks and Spatial Economics*(11), 83-99.
- OpenStreetMap. (2021a). Retrieved January 2021, <https://www.openstreetmap.org/relation/8398124#map=16/40.7632/-73.9837>.
- OpenStreetMap (2021b). *OSRM API Documentation*. Retrieved February 2021, <http://project-osrm.org/docs/v5.5.1/api>.
- Pyomo (2021). Retrieved February 2021, <http://www.pyomo.org>.
- Rushton, A., Croucher, P., & Baker, P. (2017). *The Handbook of Logistics & Distribution Management* (6. ed.). London et al.: KoganPage.
- Statistisches Bundesamt (2020). *Alle politisch selbständigen Gemeinden mit ausgewählten Merkmalen am 31.12.2020 (4. Quartal)*. Retrieved February 2021, <https://www.destatis.de/DE/Themen/Laender-Regionen/Regionales/Gemeindeverzeichnis/Administrativ/Archiv/GVAuszugQ/AuszugGV4QAktuell.html>
- Steglich, M., Feige, D., & Klaus, P. (2016). *Logistik-Entscheidungen: Modellbasierte Entscheidungsunterstützung in der Logistik mit LogisticsLab* (2nd completely revised and expanded edition ed.). Berlin et al.: De Gruyter Studium.
- Sweeney, D. C., Campbell, J. F., & Mundy, R. (2010). Teaching Supply Chain and Logistics Management Through Commercial Software. *The International Journal of Logistics Management*, 21(2), 293-308.
- Toolbox (2013). Retrieved February 2021, <https://tu-dresden.de/bu/wirtschaft/bwl/log/studium/materialien/toolbox>.
- Williams, H. (2013). *Model Building in Mathematical Programming* (5. ed.). Chichester: Wiley.
- Winkels, H.-M. (2012). *Modellbasiertes Logistikmanagement mit Excel: Lösungen von Problemen in der Logistik unter Verwendung der Tabellenkalkulation*. Hamburg: DVV.