Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia CIRP 7 (2013) 127 – 132

www.elsevier.com/locate/procedia

Forty Sixth CIRP Conference on Manufacturing Systems 2013

Milkrun Vehicle Routing Approach for Shop-floor Logistics

Dávid Gyulai[a,c]*, András Pfeiffer[a], Thomas Sobottka[b], József Váncza[a,c]

[a]*Fraunhofer Project Center at Computer and Automation Research Institute (SZTAKI), Kende str. 13-17, Budapest 1111, Hungary*
[b]*Fraunhofer Austria Research GmbH, Theresianumgasse 7, Wien 1040, Austria*
[c]*Dept. of Manufacturing Science and Technology, Budapest University of Technology and Economics, Egry J. str. 1, Budapest 1111, Hungary*
* Corresponding author. Tel.: +36-1-279-6176; fax: +36-1-466-7503, *E-mail address*: david.gyulai@sztaki.mta.hu

Abstract

In large-scale shop floors and manufacturing environment, different transportation systems are applied in order to satisfy the material requirements of the systems. The limited capacity of vehicles and time consumption of the logistics processes require effective vehicle routing approaches so as to support production without glitches. The paper gives an overview of the appropriate models and the most efficient solver algorithms of the vehicle routing problem (VRP), introduces a novel approach that uses a novel initial solution generation heuristics, and presents a local search method to solve the VRP. In order to demonstrate the capabilities of the solution proposed, the implemented software concentrates on the main industrial requirements like quick response, effective layout definition and order handling. A specific layout representation scheme is proposed which ensures interoperability between different factory-, and shop-floor planning software products.

Selection and peer-review under responsibility of Professor Pedro Filipe do Carmo Cunha

Keywords: Milkrun, Vehicle routing problem, Shop-floor logistics, Factory planning

1. Introduction

The material supply logistics in large- and medium-scale production systems has a key role in balancing the workload and ensuring a continuous and smooth running of production. The actual shop-floor logistics system influences not only performance of the production control, but even affects directly both order management and production system. In order to manage effectively both the in- and outgoing material supply system, practical transport logistics and distribution problems are usually formulated as vehicle routing problems, whose objective is to obtain a minimum-cost route plan serving a set of customers with known demands [1].

In several sectors – especially in the automotive industry – lean techniques are successfully applied to increase the productivity and keep the operation and investment costs on the lowest possible level. The generic purpose of the lean approach is to eliminate the wastes on all the levels of the production hierarchy, typically by reducing the non-value adding activities in the supply chain. In order to keep the in-line and shop-floor stocks on the lowest level, some novel logistics concepts, as for example, milkrun systems are applied. A milkrun is a manually operated, cyclic transport system delivering raw materials and finished goods, using a fixed route and time schedule [2].

According to this characterization, milkrun schedule is considered as a special vehicle routing problem (VRP) with time windows and a limited number of vehicles. In the paper a novel, multi-level planning approach is proposed which combines the advantages of existing algorithms such as local search for vehicle routing.

2. Factory logistics and vehicle routing

2.1. Lean shop-floor logistics planning

Considering lean logistics planning methodologies, overall waste reduction has become a major planning policy [3]: of the seven waste categories, eliminating unnecessary transports, waiting times, superfluous movement and excess inventory are the most relevant in lean logistics. Klug identifies the following characteristics of lean logistics [4][5]: synchronicity and a clocked

2212-8271 © 2013 The Authors. Published by Elsevier B.V.
Selection and peer-review under responsibility of Professor Pedro Filipe do Carmo Cunha
doi:10.1016/j.procir.2013.05.022

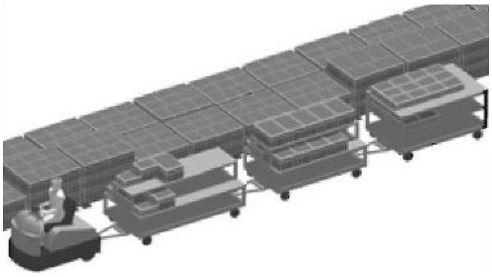

Fig. 1. Tugger train for milkrun transport

material flow, flow orientation, standardization, pull principle, stability, integration, as well as perfection. Regarding pull principle, as the most relevant feature, material is only transferred to a workstation if it is about to process the material thus reducing inventory levels by reducing shop floor stock to safety buffers only.

While in the general external logistics context these planning principles are reflected in transport concepts such as the just-in-time delivery, there are also corresponding design concepts in production and shop floor logistics. For the supplier/source side, establishing decentralized incoming goods zones and supermarkets in the vicinity of assembly stations is a means of reducing transport lengths. On the receiving side, standardized shelves and reusable containers, replenishment schemes like kanban are applied. The goal of the lean transport system between supply and workstations is the reduction of inventory and avoiding erratic peaks in production (bullwhip effect) [6]; therefore tugger trains (Figure 1) with fixed time intervals and standardized capacity and lot sizes are preferred over forklift transport.

This transport concept of scheduled tugger trains in the context of a lean-oriented logistics system is usually combined with the milkrun transport concept that is in the focus of this paper. Especially in production environments with a high product complexity, variety amid a complex and changing material flow, planning a system of milkrun transports is a complex task. An efficient system of milkruns is able to increase significantly the efficiency of the overall production logistics.

2.2. Vehicle routing approaches

Vehicle routing is a hard combinatorial optimization problem, introduced first by Dantzig and Ramser in 1959 [7][8]. The general form of VRP is the travelling salesman problem (TSP), where the objective is to find the shortest tour through a set of cities, visiting each city only once, and returning to the start point. While TSP captures the most common routing problem, the objective of the VRP is not only to find the optimal route, but also to assign the items to vehicles that ship them from one node to another [9].

The general type of VRP is the Capacitated Vehicle Routing Problem (CVRP), were the aim is to satisfy the needs of all the customers at different locations by having a given number of vehicles with capacity constraints. In case milkrun is applied as a material providing system, the time constraint of the runs has also to be added to the general problem (VRP with Time Windows – VRPTW). The general objective is to minimize the costs either by minimizing the total distance travelled, or minimizing the number of vehicles applied. A specific type of VRP is the Pickup and Delivery Problem (PDP) where the vehicles not only provide the locations with materials, but also pick up materials at the stations and deliver them to another one. The milkrun problem is also a PDP, with time windows and cyclic service [10][11].

Several approaches from the operations sciences are used to solve VRPs, as for example constraint programming (CP) that manages flexibly the various specific constraints of the VRP. Local search is one of the fundamental paradigms to find solutions for hard computational – including constraint satisfaction – problems. The basic idea underlying local search is to start with a randomly or heuristically generated candidate solution of a given problem instance, which may be infeasible, suboptimal or incomplete, and to iteratively improve this candidate solution by means of typically minor modifications. In order to avoid stagnating of the process at an unsatisfactory candidate solution, metaheuristics are applied, such as simulated annealing, tabu-search or ant colony optimization [12].

Generating the initial solution is always a crucial point when applying local search, since it affects significantly the running time of the algorithm. In order to improve the generation of the initial solutions, various existing heuristics can be applied, such as sequential insertion heuristics which simply add the following node to the existing tour based on its location. More sophisticated heuristics are the two-phased ones, which decompose the problem into a clustering- and a routing problem, with possible feedback loops between the two stages. Cluster-First Route-Second (CFRS) algorithms perform a single clustering of the vertex set and then determine a vehicle route on each cluster. The best-known CFRS algorithm is the Fisher-Jaikumar algorithm which solves the General Assignment Problem (GAP) to form the clusters [13][14].

The local search starts from the initial solution and subsequently moves from the present solution to a neighboring solution in the search space where each solution has only a relatively small number of feasible neighbor solutions and each of the movements is determined by neighborhood's operators [15]. Generally, the operators have two main groups according to the scope: inter-route and intra-route-operators. Intra-route operators are used for shortening the length of a related path,

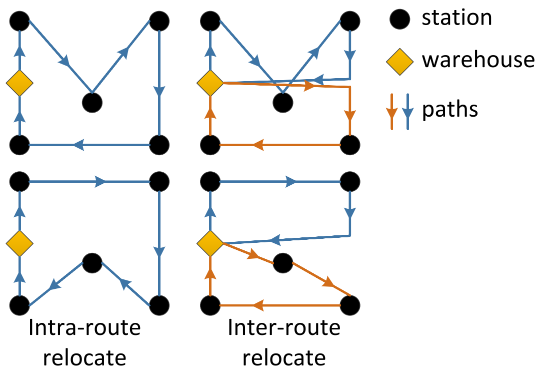

Fig. 2. Inter- and intra-route relocate operators

by contrast with inter-route operators which are able to influence the number of the required vehicles as well via affecting two routes (Figure 2). When applying local search, the combination of the neighborhood functions produces the next local optimum solution.

Although the most general forms of the VRP can be solved effectively with the above approaches, the milkrun planning problem is hard to interpret as a simple graph-search problem, and the general initial solution heuristics are difficult to transform for this problem. The mapping process of the factory layout also requires special processing; therefore general solution methods cannot be applied easily for such a special problem.

3. Statement of the problem

In what follows we define our specific milkrun planning problem. Consider a layout of the shop-floor which is defined by a set of routes and stations. The routes can be either one- or two-way ones, and each route endpoint is defined as a "routenode"; the stations belong to routes, and are placed uniformly between two routenodes (Figure 3). The problem also deals with demands that belong to the stations and defined by the amount of the transported goods and the required cycle-time of the transportation. In our case, all the demands are given in standard units, and each of them has the same cycle-time. The goods are transported by vehicles (practically by tugger trains), which are defined by their capacity and average speed.

In order to plan feasible milkruns, real-world constraints have to be considered, such as the capacity constraints of the vehicles which limit the maximum number of the transported goods. Another constraint is the time limit of the plan, which means the total time consumption of a milkrun plan cannot exceed the cycle-time of the demands. The loading and unloading points can be reached from both sides of the tugger train, and the train can approach the station from both endpoints of the route that the station belongs to.

A milkrun plan is built-up by tours, where each tour is given as the list of the visited stations and the list of the routes that the vehicle passes along. The plan does not partition the demands, and a visit of a particular station only occurs in a specific tour. A milkrun plan is characterized by its total time consumption, which is required by the vehicle(s) to perform all the tours. We consider a plan better than the other one only if its total time consumption is smaller, while it satisfies all the demands. The goal of milkrun planning is to minimize the number of the required vehicles via minimizing the time consumption of the plan.

4. Feasible milkrun planning

4.1. Shop-floor constraints and general layout representation requirements

Several possible ways exist to formalize the VRP models, depending on the different constraints of the problem and the solution algorithm. Even though there are many effective approaches to provide acceptable solution for VRP, applying them in real industrial cases can be difficult. When formalizing a VRP, most of the algorithms and approaches consider the layout of the shop-floor as a directed graph G, where a set of vertices $V=\{v_1,\ldots,v_n\}$ denotes the stations, an edge set $A=\{(r,s):r,s\in V\}$ denotes the paths between the stations, while $d_{r,s}$ is the distance between nodes $(r,s)\in A$ [16]. In many cases, this representation cannot support effectively the planning processes with feasible results, unless the application of the following constraints. Frequently, when representing a shop-floor by directed graphs, it turns out that the structure of the layout results some direct and indirect routing constraints. A direct case means that the additional equations can be formalized immediately when constructing the graph,

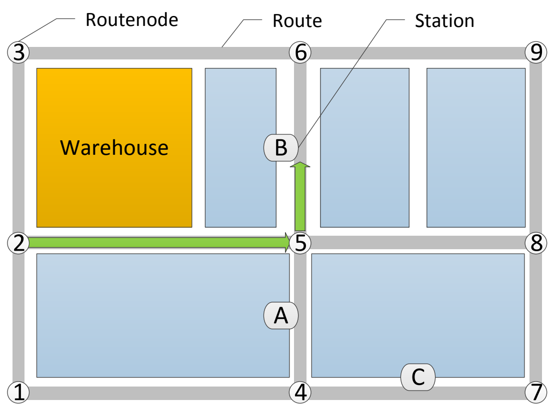

Fig. 3. Layout elements and vehicle movement

such as the asymmetric edge formulas ($d_{r,s} \neq d_{s,r}$), while the others require further consideration. Indirect constraints are usually implied by the narrow and one-way corridors, which limit the abilities of the vehicles from going to a station directly from another, even if they are connected by a path. The reason for this is shown on Figure 3, where the milkrun serving vehicle is a common tugger train. Due to the length of the vehicle, its turning abilities are limited; in case the vehicle is coming from node "2" through "5", and first serving station "B", it cannot go to the next station "A" without taking a detour in direction of "6" and "8" etc. (Figure 3).

In order to handle the shop-floor constraints effectively and to be able to avoid impossible movements of the vehicle, a novel hierarchical layout representation is proposed. The layout is defined by three main different classes having some specific attributes. This structure can be handled dynamically during the path calculations, and the asymmetric nature of the distance matrix D is represented together with the limitations of vehicles movements. Figure 4 shows the hierarchy of the layout, defined by "routenodes", "routes" and "stations". The elements of the VRP's mathematical model can be generated directly from the classes, since V is the set of the "Stations" and A is the set of the "Routes".

To calculate the distance matrix of the stations Dijkstra's shortest path algorithm was applied, which solves the single-source shortest path problem in logarithmic running time [17]. In this case, the input of the algorithm is a directed graph where the vertices are the set of the routenodes and the arcs are the routes. This graph representation is able to handle all the nodes of the layout with their connections, and by applying Dijkstra's algorithm the asymmetric $D = [d_{s,r}]$ matrix can be calculated.

4.2. Initial solution generation

In order to plan feasible milkrun cycles with the lowest operational costs, the shop-floor constraints have to be taken into consideration. To handle the constraints effectively, a so-called triplet solution can be applied which helps to avoid impossible movements of the vehicle. A triplet means that the vehicle class inside the algorithm has three main properties concerning the route and the position of the vehicle. These properties are the "current position", "previous position" and the "next position". The next position is calculated by applying the shortest path algorithm, and the previous position is prohibited for the vehicle to move on next. The triplet solutions are hidden in the background; the user cannot handle them but only the algorithms behind the interface.

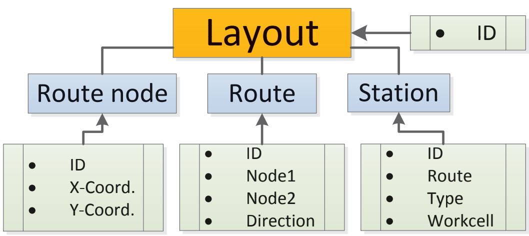

Fig. 4. Hierarchy of the layout representation

The implemented algorithm strongly focuses on the industrial requirements of vehicle routing; therefore a novel initial solution generation heuristics was implemented. The goal is to generate a solution that is as close to the criterion as possible. The heuristics introduced in section 2.2 are very efficient if we consider the classical mathematical VRP with simple graph-represented layouts.

In case the problem includes the physical constraints of the movements (e.g. by applying triplets), a novel efficient heuristics can be proposed, which is based on some practical and empirical facts only. The heuristics can be classified as a CFRS type rule, and differs from the Fisher-Jaikumar algorithm by the cluster generation method, since it does not require solving time-consuming GAP, but defines the clusters based on practical reasons.

A simple point-to-point routing algorithm calculates a path to each station applying the previously calculated D matrix. In this step, all the stations are enlisted which are passed by the serving vehicle nearby. Then a greedy search algorithm is applied to calculate the next node which must be visited by the vehicle before returning to the depot, so as to avoid violating the turnaround constraints of the vehicle. The greedy search iterates forward the nodes applying a best-first search strategy [18], and finds the first node from which the shortest path to the depot does not contain the previous node. The "roundtrip" planning method is applied for each station, and detects all the feasible rounds on the shop floor. In order to determine the set of paths for the initial solution, the paths with most visits are selected one by one while the set of unsatisfied demands is not void. Performing the services required by these cycles provides a feasible and acceptable initial solution for the planning algorithm.

4.3. Local search strategy and neighborhood functions

In order to improve the generated initial solution, neighborhood functions are applied. To be able to reach all the points of the search space, both intra- and inter-route operators are necessary to calculate with. In this case, the effect of the inter-route functions is more significant since the initial solution heuristics could provide paths with stations in a rather good sequence. The algorithm starts local search with one vehicle, and applies

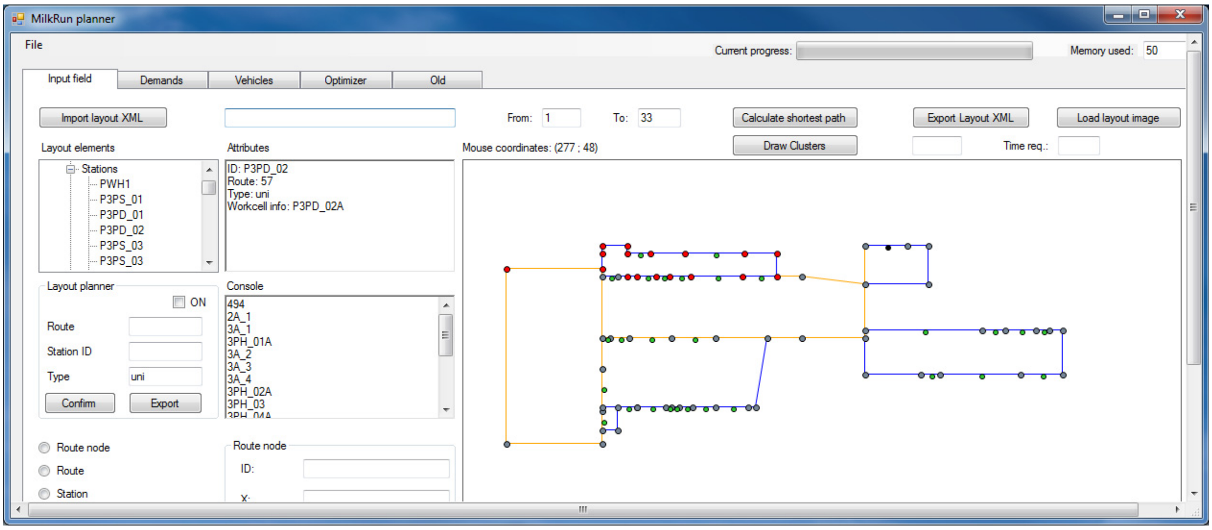

Fig. 5. The user interface of the milkrun planner application

neighborhood functions (especially "Exchange") for each combination of visits. In case the total time of the plan is greater than the required cycle-time of the milkruns, the number of vehicles is incremented, and the local search is applied again.

5. Industrial case-study and test results

The milkrun planner software which includes both the user interface and the solver algorithms were implemented in a .NET environment using C# language. Within the performance evaluation of the implemented software, a real production environment was modeled in order to analyze its capabilities. The purpose of the implemented application is to offer user-friendly and efficient milkrun planning environment; therefore a graphical, point-and-click layout definition interface and XML communication have been applied (Figure 5).

First the input parameters are processed and stored in the memory, and Dijkstra's algorithm is applied to calculate the shortest path between each station. The distance matrix is built up by vectors, each having two distance elements. This data structure is required by the applied triplet-based path planning method, which selects dynamically the proper distance parameter from the vector, based on the location of the previously visited station.

The initial solution generation uses the distance matrix to create the tours, based on the station loops detected on the layout. The initial solution contains only feasible paths that satisfy the time constraints as well as the capacity constraints of the actual vehicle. To avoid overloaded vehicles, a pre-check method is performed: the vehicle is loaded by all the raw materials in the depot, and then all the other loading and unloading processes are calculated station-by-station to simulate the real milkrun process. In order to calculate with time constraints, both the travel-time (with constant average speed) and the loading/unloading time (time/item) are considered. The purpose of the generated initial paths is to satisfy all the demands by the lowest possible number of feasible cycles.

The local search algorithm takes the initial paths and reduces their total time consumption by applying neighborhood functions where it is possible. The algorithm iterates through all the combinations of the stations, and detects the possible insertions from one path to another. As local search iterates through the paths in a random sequence, different test runs can result different milkrun plans. In order to approximate the global optimum solution, the implemented application performs the local search several times and selects the most appropriate schedule from the generated solutions.

The capabilities of the algorithm are tested in a real production environment, which is a large-scale automotive production system with 2 factory halls, 67 stations and 96 routenodes. The milkrun plan has to satisfy 193 various demands in 60 minutes cycle-time. The analyzed system requires really efficient planning process, since the total length of the routes is over 2500 meters that affects critically the total time of the milkrun schedule. The application of inefficient milkrun schedule results low-utilized vehicles and high number of cycles.

The goal of the test in this case is to minimize the number of vehicles required to perform the services, via minimizing the time of the cycles. The proposed initial solution heuristics generates eight feasible paths with 20400 seconds total time. First, the algorithm tries to optimize the plan by applying one vehicle, and increases the number of the vehicles only if the improvement steps

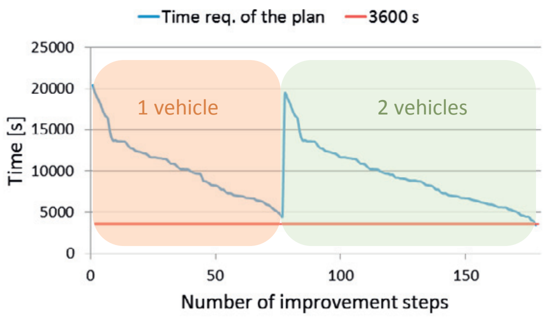

Fig. 6. Test results of the case study

cannot decrease the time requirement of the total plan under the cycle-time (3600 seconds). By this way, the test run could optimize the plan for two vehicles in 177 iterations, and generated a schedule which time requirement is only 57 minutes (Figure 6).

6. Conclusions and future work

The paper introduced a novel approach that uses an initial solution generation heuristics and a local search method to solve the milkrun planning problem. In order to demonstrate the capabilities of the solution proposed, a software prototype has been developed and tested on real-life industrial data.

Future work is primary aimed at extending the model with demand partitioning so as to increase capacity utilization of the transportation vehicles. Moreover, handling inhomogeneous demand types (physical aspects of material handling) requires further constraints to be included in the model.

Secondly future work will be dedicated to making the solution available in practical industrial applications. In an ongoing research project the milkrun planning algorithm will be implemented in a factory- and a logistics planning application currently under implementation. Those planning scenarios, which all have been encountered in the course real-life case studies, are characterized by highly complex systems of milkrun transports and call for the support of systematic, optimized planning.

Acknowledgements

The research has been partially supported by the National Office for Research and Technology (NKTH) grant "Digital, real-time enterprises and networks" (OMFB-01638/2009), as well as the „Talent care and cultivation in the scientific workshops of BME" project (TÁMOP-4.2.2.B-10/1--2010-0009.).

References

[1] Lau, H., Sim, M., Teo, K., Vehicle Routing Problem with Time Windows and a Limited Number of Vehicles, European Journal of Operational Research, 2003. Volume 148(3), 559-569.

[2] Droste, M.; Deuse, J.: A Planning Approach for In-plant Milk Run Processes to Optimize Material Provision in Assembly Systems, Proceedings of 4th CIRP CARV 2011, 605-610.

[3] Arnold, D., et al., 2008. Handbuch Logistik, 3rd Edition, Springer, Heidelberg, p. 8.

[4] Klug, F., Logistikmanagement in der Automobilindustrie, Springer, Berlin Heidelberg 2010, p. 256.

[5] Womack, J.P., Jones, D.T., Lean Thinking, Free Press, New York 2003, p. 21.

[6] Bichler, K., Krohn, R., Gabler Kompaktlexikon Logistik, 2nd Edition, Springer, Wiesbaden 2011, p. 117.

[7] Savelsbergh, M. W. P., Sol, M. The General Pickup and Delivery Problem, Transportation Science 1995, 29(1); 17-29.

[8] Dantzig, G. B., Ramser, J. H., The Truck Dispatching Problem, Management Science 1959, 6(1), 80-91.

[9] Pillac, V. et al., A Review of Dynamic Vehicle Routing Problems, European Journal of Operational Research 2013, 225(1), 1-11.

[10] Lu, Q., Dessouky, M., An Exact Algorithm for the Multiple Vehicle Pickup and Delivery Problem, Transportation Science 2012, 38, 503-514.

[11] Berbeglia, G. et al., Feasibility of the Pickup and Delivery Problem with Fixed Partial Routes: A Complexity Analysis, Transportation Science August 2012

[12] Rossi, F., Beek , van P., Walsh, T., Handbook of Constraint Programming (Foundations of Artificial Intelligence), Elsevier 2006, 245-269.

[13] http://neo.lcc.uma.es/vrp ; The VRP Web

[14] Hashimoto, H.; Studies on Local Search-Based Approaches for Vehicle Routing Problems; PhD thesis, Kyoto University 2008.

[15] Carić, T. et al., A Modelling and Optimization Framework for Real-World Vehicle Routing Problems, Vehicle Routing Problem, Tonci Caric and Hrvoje Gold (Ed.)

[16] Matai R., Singh, S. and Mittal, M. L., Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches, Traveling Salesman Problem, Theory and Applications, Donald Davendra (Ed.)

[17] Wilt, C. M., Thayer, J. T., Ruml, W., A Comparison of Greedy Search Algorithms, Proceedings of the Third Annual Symposium on Combinatorial Search (SOCS-10), 2010

[18] Russell, S. J., Norvig, P., Artificial Intelligence: A Modern Approach (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall 2002, 94-95.