A local-search-based solution for milkrun logistics optimization

DÁVID GYULAI[1,2], LÁSZLÓ MONOSTORI[1,2]
[1]Fraunhofer Project Center for Production Management and Informatics, Computer and Automation Research Institute (SZTAKI), Kende str. 13-17, Budapest, HUNGARY
david.gyulai@sztaki.mta.hu, laszlo.monostori@sztaki.mta.hu
[2]Budapest University of Technology and Economics, Dept. of Manufacturing Science and Technology, Egry J. str. 1, Budapest, HUNGARY

Abstract: In order to satisfy the material supply needs of large scale shop-floors and production systems, different logistics solutions are applied. In lean manufacturing enterprises, the material supply is pulled by the demands of manufacturing/assembly processes; therefore, a milkrun service is often applied to support the production without glitches. The milkrun logistics planning is a special case of the vehicle routing problem (VRP), and requires effective solution approach to satisfy the various constraints, and minimize the cost of the service. The paper gives an overview about the most efficient VRP solver algorithms, and proposes a novel initial solution generation heuristics which is specially focused on flexible milkrun planning. In order to demonstrate the capabilities of the solution, a software environment is introduced which concentrates on the main industrial requirements of logistics planning like effective layout definition, quick response of the delivery service and effective order handling.

Keywords: vehicle routing problem, milkrun, local search, factory logistics

1 Introduction

The material provision of large- and medium-scale production systems requires prudent planning and control, since it influences not only the performance of the production, but even affects the order management directly. In order to manage effectively the material provision of the production, practical transport logistics and distribution tasks 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 state-of-the art manufacturing systems, lean processes are successfully applied to eliminate the wastes on all the levels of production, and they raise the overall efficiency of the system by reducing the non-value adding activities in the process chain. The shop-floor logistics processes have to be adapted to the production system, therefore the waste reduction in the material provision also has a key role in balancing the workload and ensuring a smooth running of production. A well operating lean logistics system keeps the inventories and operational costs on the most cost efficient levels, via providing the materials what needed, exactly when they are needed. In order to satisfy these requirements, concepts of the lean can be applied, which are primarily aimed at controlling the transport services duly to the pull strategy.

The paper is focused on the *milkrun* service, which is a manually operated, cyclic transport system delivering raw materials (or sub-assemblies) and disposing empties based on consumption using a fixed route and time schedule [2]. According to this characterization, milkrun schedule is considered as a special vehicle routing problem with time windows and a limited number of vehicles. In the paper a novel, multi-level planning approach is pro-posed which combines the advantages of existing algorithms such as local search for vehicle routing.

First, an overview of the state-of-the-art factory logistics is given, considering the recent shop-floor characteristics and planning demands. Then the main solution approaches and algorithms are introduced, as well as the critical points of milkrun planning. In section 4, the proposed layout representation hierarchy and the novel initial solution generation heuristics method are proposed. The latter focuses on the real industrial criteria and give a feasible milkrun schedule. In the last section, an industrial case-study and some numerical results present the efficiency of the developed milkrun planning method.

2 Shop-floor logistics

2.1 Milkrun service

Klug identifies seven waste categories of lean; from which eliminating unnecessary transports, waiting times, superfluous movement and excess inventory are the most relevant ones in logistics [3]. In demand driven logistics systems, material is only transferred to a workstation if it is about to processed, to reduce shop floor stock to safety buffers only [4]. This can be realized via applying *heijunka*, which is aimed at leveling the production and controlling the just-in-time material flow processes [5]. Another effective solution of the lean logistics is the *kanban* system, which controls the production quantities of the processes [5].

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 with fixed time intervals and standardized capacity are preferred over forklift transport. Thus, milkrun service usually applies tugger train vehicles to transport the goods on the shop-floor. 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 significantly increase the efficiency of the overall production logistics.

2.2 Milkrun planning as a vehicle routing problem

Vehicle routing is a hard combinatorial optimization problem, introduced first by Dantzig and Ramser in 1959 [7][8]. The general VRP is known as 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 [9]. 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 science 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 approaches 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, and to iteratively improve that by means of defined functions [12].

Generating the initial solution is always a crucial point when applying local search, since it significantly affects the running time of the algorithm. Sophisticated heuristics for the VRP 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]. When applying local search, the combination of the neighborhood operators 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 Problem formulation

In what follows we define the 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”. Along a specific route, there can be more stations (Figure 1). The problem also considers demands that belong to the stations and defined by the amount of the transported goods in standard units and the required cycle-times of the transportation. 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 are 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 paths, where each path 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 path. A milkrun plan is characterized by its total time consumption, which is required by the vehicle(s) to perform the plan. A milkrun plan is considered to be 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 The proposed planning process

4.2. Layout representation for vehicle routing

As it was mentioned in the previous section, solving the general VRP is not enough to plan feasible milkrun schedule, since most of the routing algorithms does not calculate with the physical constraints determined by the shop-floor.

Most of the general solution approaches consider the problem as a graph-search, where the stations are represented by vertices and the routes are represented by edges [16].

In many cases, this representation does not 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, such as the asymmetric edge formulas, 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. These constraints are implied by the moving abilities of the common applied tugger train, since it cannot turn in the middle of an aisle, and in many of the cases it have to take a detour before visiting the next station (Figure 1).

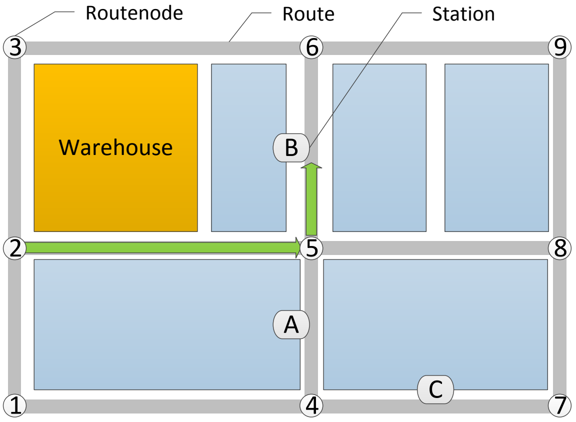


Figure 1.: Layout representation and vehicle movement

In order to handle the shop-floor constraints effectively and 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 their own specific attributes. This structure can be handled dynamically during the path calculations, and the asymmetric nature of the distance matrix is represented together with the limitations of vehicles movements. Figure 2 shows the hierarchy of the layout, defined by “routenodes”, “routes” and “stations”.

To calculate the distance matrix of the layout which store the shortest paths between the stations, Dijkstra’s algorithm was applied that 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 edges 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 distance matrix can be calculated.

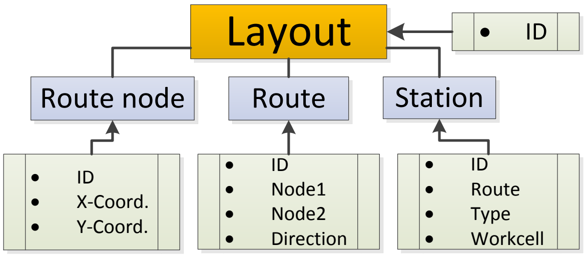


Figure 2.: Layout representation hierarchy

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, the routing algorithm needs to select dynamically the stations to be visited. Thus, a so-called triplet solution is applied which helps to avoid impossible movements of the vehicle. A triplet is consisted of three identical nodes: thee current, the previous and the next position of the vehicle. The next position is calculated dynamically when determining a particular path of the vehicle. The milkrun planning method 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 proposed heuristics can be classified as a CFRS type rule, and differs from the Fisher-Jaikumar algorithm [13] by the cluster generation method, since it does not require solving time-consuming GAP, but defines the clusters based on practical reasons.

To generate a feasible initial solution, the algorithm calculates a path to each station applying Dijkstra’s shortest path algorithm. 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 tours 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.

The pseudocode of the proposed algorithm is the following:

$v \in V$	set of stations
$p := \{v \mid v \in V\}$	path
$P\{p\}$	set of paths
$D = [d_{rs}] \mid (r,s) \in V$	distance matrix
$I\{p\}$	initial solution
q_i	demands of the ith station
$Q = \{q_i\}$	demand set of the stations
t_c	cycle time
c	capacity of the vehicle

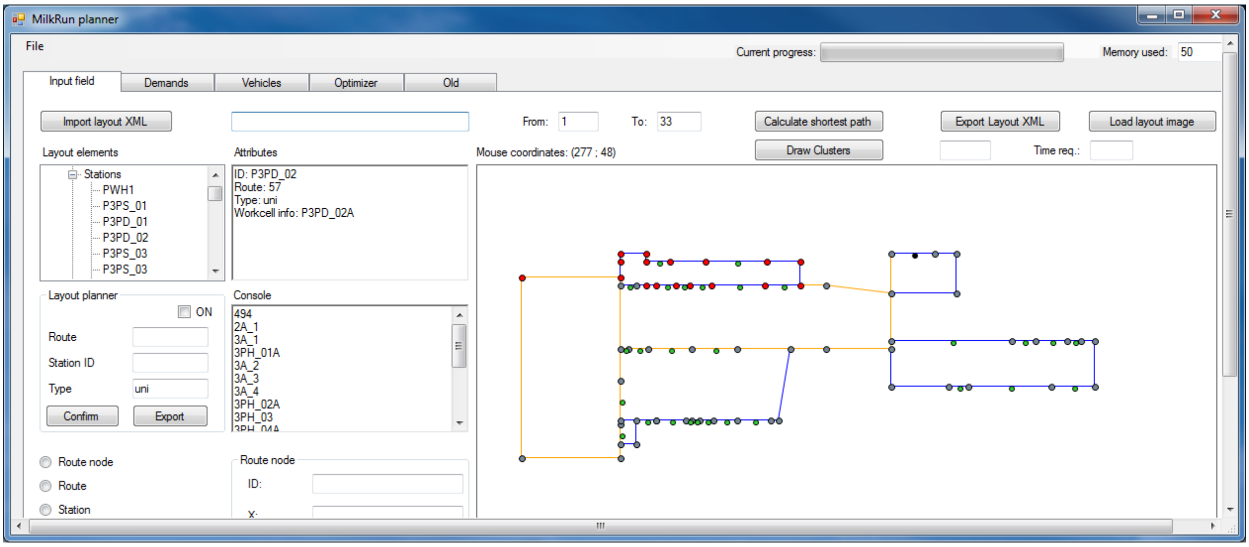
```
initialize $V, I, Q, C\{v\}, c, t_c$
foreach $i$ in $V$
    calculate $D_{0i}$
    foreach $v$ in $D_{0i}$
        add $v$ to $C_i$
    calculate $D_{i0}$
    foreach $v$ in $D_{i0}$
        add $v$ to $C_i$
align $C$ [length($C_{i+1}$) < length ($C_i$)]

while $|Q| > 0$
    while time($C_1$) > $t_c$ or load($C_1$) > $c$
        remove rand $v$ from $C_1$ ($v \neq i$)
    foreach $v$ in $C_1$
        delete $q_v$ from $Q$
    add $C_1$ to $I$
    delete $C_1$ from $C$
return $I$
```

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, although 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 performs the local search to minimize the paths, and the cycle-time of them. Then, a simple bin packing problem is considered, to allocate the paths to a minimal number of vehicles. To solve the problem, *first fit* heuristics was applied which is a proven 11/9 OPT solution algorithm [19].

5 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



Current progress:

Memory used: 50

From: 1

To: 33

Time req.:

Mouse coordinates: (277 ; 48)

Stations
- PWH1
- P3PS_01
- P3PD_01
- P3PD_02
- P3PS_03
- P3PS_03

ID: P3PD_02
Route: 57
Type: uni
Workcell info: P3PD_02A

□ ON

Route

Station ID

Type: uni

```
494
2A_1
3A_1
3PH_01A
3A_2
3A_3
3A_4
3PH_02A
3PH_03
3PH_04A
```

ID:

X:

Figure 3.: User interface of the milkrun planner application

interface and XML communication have been applied (Figure 3).

The generated initial solution for the problem contains only feasible paths that satisfy the time constraints as well as the capacity constraints of the 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 ten feasible paths with the total time of 7702 seconds. As it is seen on Figure 4, the local search method decreases the number of paths to three, with the total time consumption of 6372 seconds. As it is still over the cycle-time, a bin packing problem is solved to allocate the paths to a minimal number of vehicles. The calculated three paths can be performed by two vehicles, which means that one of the tugger trains has to return to the depot in its cycle for loading. The result of the planning is a feasible milkrun schedule, which time requirement is 3329 seconds and satisfies all the demands within the cycle-time limit.

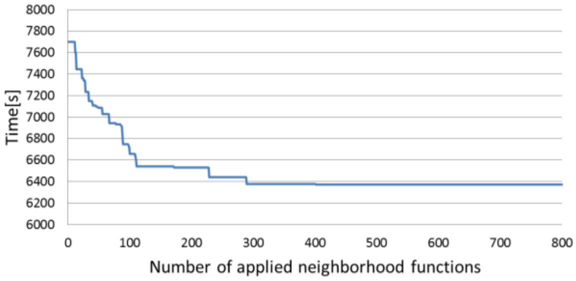


Figure 4.: Improvement of the initial solution

6 Conclusions and future work

In the first sections, an overview was given about the state-of-the art shop-floor logistics, and the general formulations of the vehicle rouing problem. In section 4, a solution was proposed that uses a novel layout representation and initial solution generation heuristics to solve the milkrun planning problem. In order to demonstrate the capabilities of the solution, 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*, Vol.148, No.3, 2003, pp. 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 International Conference on Cahngeable, Agile, Reconfigurable and Virtual Production 2011*, pp. 605-610.

[3] Klug, F., *Logistikmanagement in der Automobilindustrie*, Springer, 2010

[4] Womack, J.P., Jones, D.T., *Lean Thinking*, Free Press, 2003

[5] Lippolt, C. R., Furmans, K., Sizing of Heijunka-controlled Production Systems with Unreliable Production Processes, *Lean Business and Systems Beyond*, Vol. 257, 2008, pp. 11-19.

[6] Bichler, K., Krohn, R., *Kompaktlexikon Logistik*, Springer, 2011

[7] Savelsbergh, M. W. P., Sol, M., The General Pickup and Delivery Problem, *Transportation Science,* Vol.29, No.1, 1995, pp. 17-29.i

[8] Dantzig, G. B., Ramser, J. H., The Truck Dispatching Problem, *Management Science* Vol.6, No. 1, 1959, pp. 80-91.

[9] Pillac, V. et al., A Review of Dynamic Vehicle Routing Problems, *European Journal of Operational Research*, Vol. 225, No. 1, 2013, pp. 1-11.

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

[11] Berbeglia, G. et al., Feasibility of the Pickup and Delivery Problem with Fixed Partial Routes: A Complexity Analysis, *Transportation Science*, Vol.46, 2012, pp. 359-373.

[12] Rossi, F., Beek, van P., Walsh, T., *Handbook of Constraint Programming,* Elsevier, 2006

[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*, Hrvoje Gold, 2008

[16] Matai R., Singh, S., Mittal, M. L., Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches, *Traveling Salesman Problem, Theory and Applications*, InTech, 2010

[17] Wilt, C. M., Thayer, J. T., Ruml, W., A Comparison of Greedy Search Algorithms, *Proceedings of the Third Annual Symposium on Combinatorial Search*, 2010, pp. 129-136.

[18] Russell, S. J., Norvig, P., *Artificial Intelligence: A Modern Approach (2nd ed.)*, Prentice Hall, 2002

[19] Yue, M., A simple proof of the inequality FFD (L) $\leq$ 11/9 OPT (L) + 1, $\forall$ L for the FFD bin-packing algorithm, *Acta Mathematicae Applicatae Sinica*, Vol.7, No.4, 1991, pp. 321-331.