Milk-run routing and scheduling subject to different pick-up/delivery profiles and congestion-avoidance constraints

Bocewicz, Grzegorz; Nielsen, Peter; Zbigniew, Banaszak

Published in:
IFAC-PapersOnLine

DOI (link to publication from Publisher):
10.1016/j.ifacol.2019.08.089

Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 52-8 (2019) 313–320

Milk-run routing and scheduling subject to different pick-up/delivery profiles and congestion-avoidance constraints

Grzegorz Bocewicz,* Peter Nielsen** and Banaszak Zbigniew*

* *Dept. of Computer Science and Management, Koszalin University of Technology, Poland, bocewicz@ie.tu.koszalin.pl, Zbigniew.Banaszak@tu.koszalin.pl*

*** Dept. of Materials and Production, Aalborg University, Aalborg, Denmark, peter@mp.aau.dk*

Abstract: Two kinds of intertwined decisions: the routing decisions, which determine the set of sequences of stations visited by each tugger train's route, and the scheduling decisions, which plan congestion-free movements of tugger train fleets, are considered. The problem under study can be seen as extension of the pick-up and delivery problem with time windows in which different profiles of separately executed delivery and pick-up operations are assumed. The NP-hard character of the problem considered follows from its roots derived from the vehicle routing and the deadlock-avoidance problems. In this regard, a constraint programming paradigm allowing the further integration of multi-period, multi-trip and multi-commodity flows with various customers' demands as well as distribution network topology constraints is applied. Consequently, a recursive formulation of a well-known constraint satisfaction problem is proposed. The computer experiments provided illustrate the possibility of using the approach presented in systems of real-life scale.

Keywords: Milk-run system, vehicle routing problem, congestion avoidance, constraint programming.

1. INTRODUCTION

A milk-run concept applied in a material-providing setup can be defined as a manually operated, cyclic transport system delivering raw materials and finished goods, using a fixed route and time schedule (Droste and Deuse 2011; Patel et al. 2014). Taking into account a limited number of vehicles (logistics trains) and their capacity constraints as well as the time windows constraining execution of the pick-up and delivery operations, a corresponding milk-run planning problem can be seen as a special case of a well-known hard combinatorial optimization vehicle routing problem (Perronnet et al. 2014). The goal of milk-run planning (i.e. determining routes and schedules) is to minimize the number of the tugger trains required to perform the services, through minimizing the time of the cycles (Gyulaia et al. 2013).

Concurrent operation of multiple vehicles in a limited structure of distribution network layout results in a high probability of occurrences of congestion involving effects including deadlocks, livelocks, collisions, overcrowdings and so on. Consequently, since congestion occurrences are implied by resource conflicts caused by milk-run flows demanding access to common shared resources (e.g. segments of routing paths) the relevant traffic control policy guaranteeing congestion-avoidance routing plays a pivotal role. Typically, a problem of vehicles conflict-free routing through a given layout of distribution network belongs to a class of NP-hard problems (Lenstra and Rinnooy Kan 1981).

In this context, our study concerns finding a computationally effective approach aimed at simultaneous routing and scheduling of logistics trains' flow as well as a design of distribution network infrastructure. This means that a reference model is sought which allows the formulation of a decision problem that captures the importance of striking an equilibrium between potential expectations regarding milk-run traffic and the capacity of the existing distribution network, focusing on resolving resource conflicts, i.e. conflicts that arise when different activities simultaneously request access to the common shared resources (e.g. intersections and/or guideway line segments) of limited quantity. Due to the different character of decision variables (strings, sets, integers) and linking them to various specific constraints (e.g. logical, algebraic, set theory), implementation of a declarative modeling framework seems to be well suited. It should be recalled that in declarative models the focus is on what the solution should be. In other words, in contrast to imperative approach models of computations, which are expressed in terms of states and sequences of state-changing operations, taking an "inside-out" approach simply describes how a solution is obtained, declarative models take an "outside-in" approach. Instead of specifically describing how the process has to work, only the essential characteristics are described. Therefore, constraint programming is employed, primarily because of its capability of fast constraint satisfaction problems prototyping. The provided illustrative example shows the possibility of using the approach presented in the DSS aimed at milk-run fleet planning subject to different profiles of separately executed pick-up and delivery processes and congestion-avoidance constraints. By a profile of the pick-up or delivery process we mean an outline of its main characteristics such as the pass route, the cycle, operation times and so on. In this context, two cyclic processes composed of events occurring with the same frequency, although of different operation times, are treated as having different profiles. The results fall within the scope of the research reported in our previous papers (Bocewicz et al. 2017, 2019).

2405-8963 © 2019, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2019.08.089

The remainder of the paper is organized as follows. Section 2 reviews the literature. An example of a milk-run fleet routing and scheduling problem setting up a proposed methodology is described in Section 3. In Section 4 a mathematical formulation of the problem, the proposed solution and the computational results are reported and analyzed. Finally, conclusions and future works are considered in Section 5.

2. RELATED WORK

In a milk-run system, routes, time schedules, type and number of parts (components) to be transported are assigned to different logistics trains so that they can collect/deliver orders from/to different suppliers/customers. The benefits of using a system of this type include the improved efficiency of the overall logistics system and potential substantial savings in shortening the total distance travelled and minimizing the number of vehicles applied, along with remarkable cost advantages related to inventory costs (Patel et al. 2014). In this context, the milk-run driven the in- and outgoing material supply and distribution problems are usually recognized and formulated as vehicle routing problems (VRP), whose objective is to obtain a minimum-cost route plan serving a set of customers with known demands, i.e. to assign the items to vehicles that ship them from one depot to another (Lau et al. 2003; Pillac et al. 2013). Consequently, the milk-run driven problems of components'/parts'/commodities' distribution can be classified similarly to extensively studied extensions of VRP concerning, for instance:

- Capacitated VRP, where the aim is to satisfy the needs of all the customers at different locations by having a given number of vehicles with capacity constraints.
- Consistent VRP, in which the same customers are serviced by the same driver at roughly the same time period over the planning horizon vehicle routing problem with time windows, which is a generalization of the VRP where the service of any customer starts within a given time interval, called a time window.
- Pick-up and Delivery Problem VRP, where the vehicles not only provide the locations with materials, but also pick up materials at the stations and deliver them to others.
- VRP with time windows, which is a generalization of the VRP where the service of any customer starts within a given time interval, called a time window (Lau et al. 2003; Suprayogi et al. 2009),
- VRP with Backhauls, also known as the linehaul-backhaul problem, is an extension of the VRP involving both delivery and pick-up points (Ong and Suprayogi 2011),
- VRP with multi-trip multi-traffic pick-up and delivery problem with time windows and synchronization is a combination of variants of the vehicle routing problem with multiple trips, vehicle routing problem with time window, and vehicle routing problem with pick-up delivery (Suprayogi et al. 2009).

Besides a huge volume of papers covering different technical problems and following issues derived from the everyday life practice there is a large volume of methods and problem-solving techniques employed in the course of their modelling and investigation. The modelling frameworks consist of operation research methods (such as linear and nonlinear programming, MLP, simulation, and so on) and artificial intelligence methods, such as evolutionary computation (including metaheuristic and stochastic optimization algorithms) (Gyulaia et al. 2013; Nguyen et al. 2017), and fuzzy-set methods. In the context of the abovementioned scope of problems and methods addressed to logistic trains' routing and scheduling only a limited number of papers are devoted to robust and congestion-free scheduling of a fleet of vehicles subject to in-plant layout constraint. In this respect, the most relevant are factors dependent on critical and often unpredictable traffic congestions which occur when logistics operators allocate too many collecting tasks to the available vehicles, generating unperformed activities due to assumed just-in-time constraints imposed by time windows of customer services (Nguyen et al. 2017; Lau et al. 2003). The majority of the research in the field of distribution logistics is devoted to the analysis of the methods of organizing transport processes in ways that minimize the size of the fleet, the distance travelled (energy consumed), or the space occupied by a distribution system. In focusing on the search for optimal solutions, these studies implicitly assume that there exist admissible solutions, e.g. ones that ensure the collision- and/or deadlock-free (congestion-free) flow of concurrent transport processes. In practice, this requires either online updating (revision) of the routing policies used, or prior (offline) planning of congestion-free vehicle routes and schedules. Studies on generating dynamic routing policies are conducted sporadically; even less frequent are investigations of the robust routing and scheduling of milk-run traffic, which are, by and large, limited to AGV systems. This is due to the fact that the congestion-avoidance problem, which conditions the existence of admissible solutions, is an NP-hard problem (Wysk et al. 1994). Consequently, in real-life applications, congestion-avoidance methods (e.g. deadlock prevention) are used, which implement sufficient conditions for the collision-free execution of processes. Methods that are most commonly used for such purposes include those that use the formalism of max-plus algebra (Polak et al. 2004) and constraint programming (Sitek and Wikarek 2017). It should be noted, however, that the possibility of the fast implementation of the process-synchronization mechanism (e.g. employed by deadlock-prevention methods) comes at the expense of omitting some of the potentially possible (e.g. including optimal solutions) scenarios for deadlock-free execution of the processes. The shortcomings of methods providing admissible solutions restrict their implementation in DSS systems, in particular in systems supporting planning of milk-run traffic flows. Given this background, our contribution boils down to the assessment of the possibility of using declarative modelling in decision support tools dedicated for prototyping in-plant milk-run traffic systems.

3. MILK-RUN FLEET ROUTING AND SCHEDULING

Consider the milk-run system layout which is shown in Fig. 1. The fleet composed of two trains TT_1, TT_2 (marked in orange and green, respectively, see Fig. 1) is used to service seven production cells, the supermarket and the warehouse while providing dedicated material pick-up/delivery operations. Train TT_1 following transportation route designated by docking stations $M_1 - M_5$ delivers intermediate components from the supermarket (M_8) to the relevant

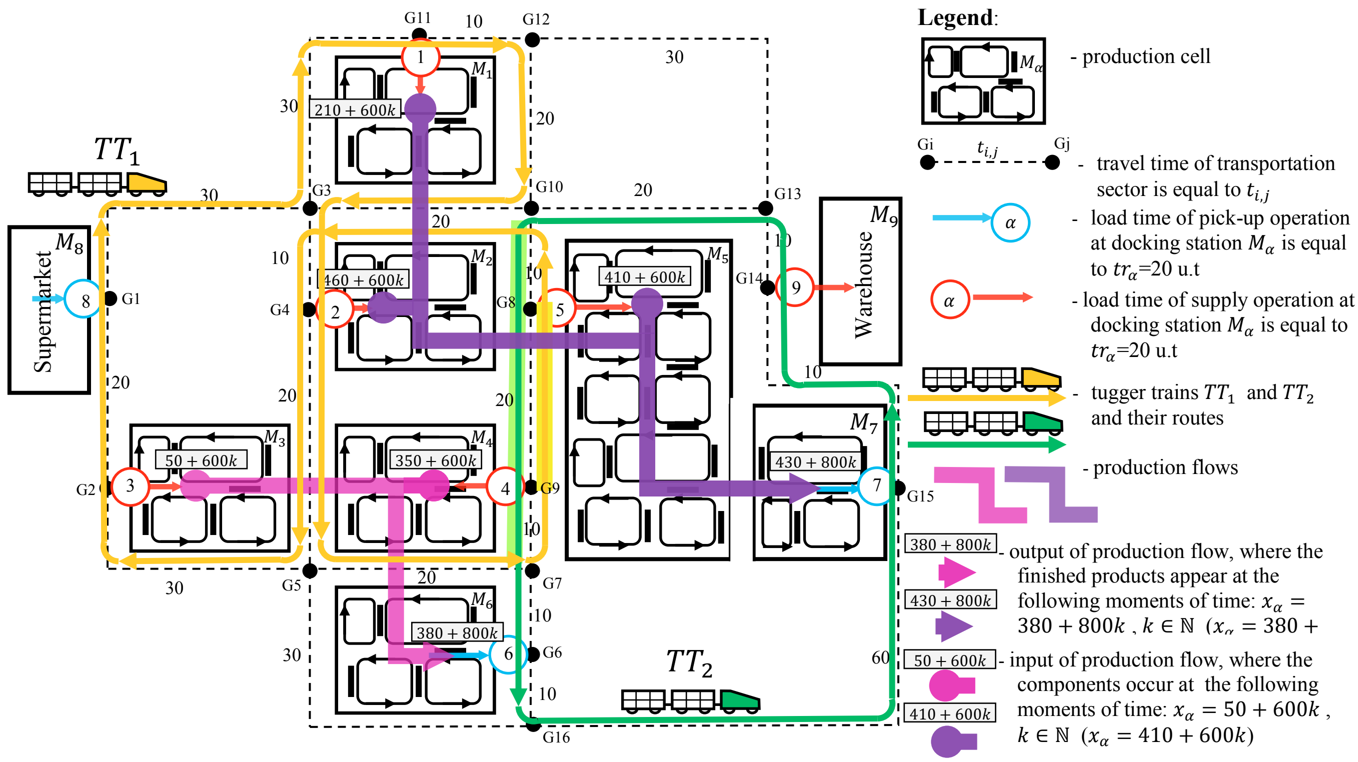

Fig. 1 Layout of the milk-run transportation system

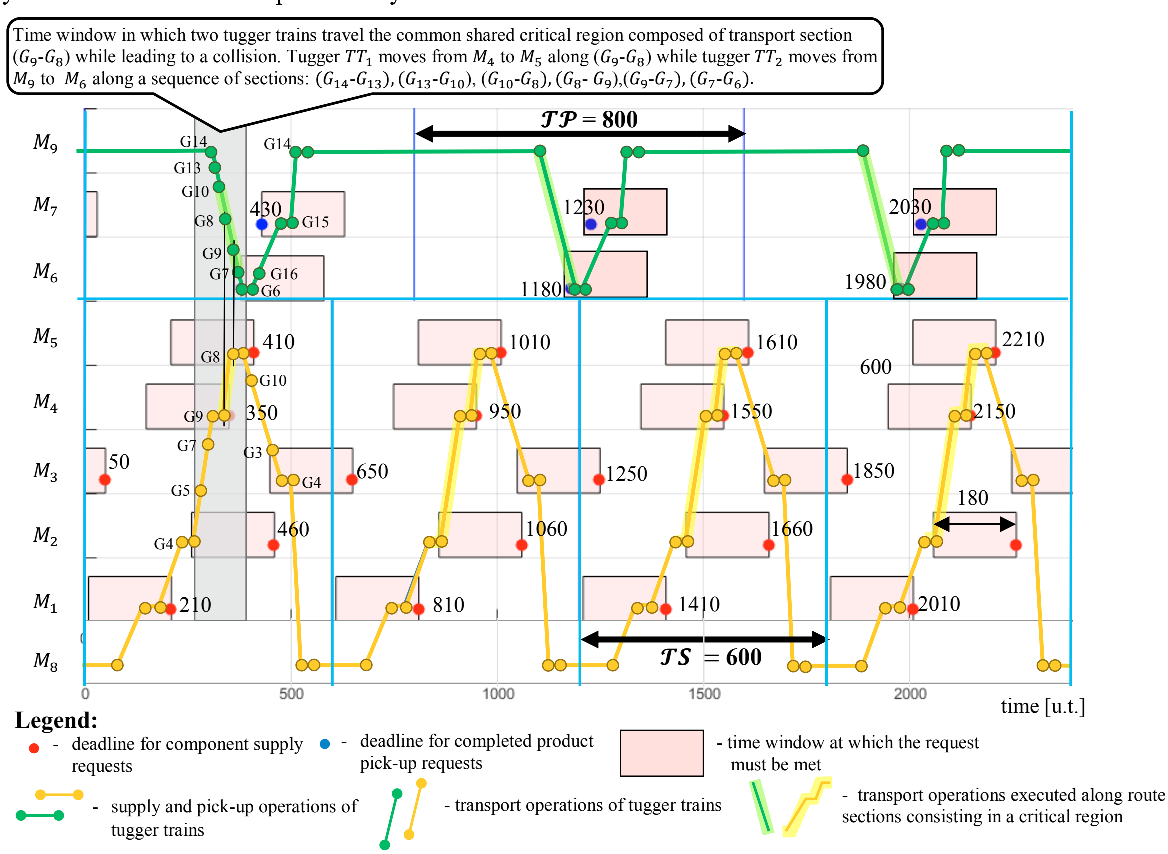

Fig. 2 The cyclic schedule of tugger trains TT_1,TT_2 routings following streams of supply and pick-up requests

production cells. In turn, the tugger train TT_2 following the transportation route designated by docking stations M_6, M_7 picks up completed goods to supply them to the warehouse (station M_9) from the relevant set of production cells. Delivered parts (components) and products assembled from them are processed along two separated production flows marked in violet and magenta (see Fig. 1). Takt times of considered production flows determine the timetables of supply and pick-up operations serviced at each production cell. Consequently, the supply order (requested by M_1–M_5) and pick-up requests (serviced by M_6, M_7) appear cyclically (within a given time windows) with two different periods: 600 u.t. (units time) and 800 u.t., respectively (Fig. 2). All requests must be met within a given time window sizes of 180 u.t. This means the supply operations should be completed within the period of 180 u.t before request moments, and the pick-up operations should be started within the period of 180 u.t. after request moments.

Consequently, the supply orders and pick-up requests can be seen as interacting, however independently flowing streams of deliveries servicing both parts' supply and products' pick-up. Assuming that:

- the train TT_1 is used to transport components from supermarket to the docking stations M_1 –M_5 (cyclically with the period 600 u.t.)
- the train TT_2 is used to transport products to the warehouse from docking stations M_6, M_7 (cyclically with the period 800 u.t.)
- the travel times along transportation sections are the same for each train and known in advance (see Fig. 1),
- the times of loading and unloading operations are the same for each docking station (see Fig. 1)
- at a given moment a docking station (transport section) can be occupied by a unique train only
- a resource (e.g. a warehouse, a supermarket, a production cell) is serviced by a unique docking station M_α,
- each request should be satisfied in a given time window (see Fig. 2)

the following question can be considered:

Do routes exist for the given fleet of tugger trains such that items supply/picked-up along them to and from the given docking stations due to time windows are determined by assumed streams of requests?

4. DECLARATIVE MODELLING

4.1 Problem statement

The mathematical formulation of the model considered employs the following:

Symbols:

TT_v: v-th transport process (performed by the v-th train);

M_α: α-th docking station (associated with the warehouse, the supermarket, the production cell);

o_α: α-th delivery operation (operation of supply/pick-up of a part/product to/from a docking station);

b_α: index of supply operation which precedes o_α;

f_α: index of supply operation which follows o_α;

Parameters:

l: number of transport means;

ω: number of docking stations;

x_α: moment of request access to the docking station M_α,

tr_α: operation time of o_α,

$d_{\alpha,\beta}$: travel time between the docking station M_α and the docking station M_β;

$\mathcal{TP}$: assumed value of production takt time for pick-up requests.

$\mathcal{TS}$: assumed value of production takt time for supply requests.

$\mathcal{K}_{\varepsilon\beta-\lambda\gamma}$: intersection indicator. $\mathcal{K}_{\varepsilon\beta-\lambda\gamma} = 1$ if (o_ε, o_β) and (o_λ, o_γ) are pairs of consecutive operations, and the path connecting docking stations M_ε, M_β, are executed crosses the path that connects the docking stations M_λ, M_γ; $\mathcal{K}_{\varepsilon\beta-\lambda\gamma} = 0$ in the remaining cases.

Sets and sequences:

TT: set of transport means TT_v (tugger trains);

M: set of docking stations M_α;

$\mathcal{O}$: set of delivery operations o_α;

$\mathcal{P}$: set of pick-up operations, $\mathcal{P} \subseteq \mathcal{O}$;

$\mathcal{S}$: set of supply operations, $\mathcal{S} \subseteq \mathcal{O}$;

B: sequence of predecessor indices of delivery operations, $B = (b_1, \dots, b_\alpha, \dots, b_\omega)$, $b_\alpha \in \{0, \dots, \omega\}$;

F: sequence of successor indices of delivery operations, $F = (f_1, \dots, f_\alpha, \dots, f_\omega)$, $f_\alpha \in \{1, \dots, \omega\}$

$\pi_v = \left(M_{v_1}, \dots, M_{v_i}, M_{v_{i+1}}, \dots, M_{v_\mu}\right)$where: $v_{i+1} = f_{v_i}$ for $i = 1..\mu - 1$ and $v_1 = f_{v_\mu}$: route of v-th transport process following sequence (determined by F) of docking stations serviced by train TT_v; .

Π: set of routes π_v;

Variables:

xt_α: start time of operation o_α, on the α-th docking station

yt_α: end time of operation o_α

xs_α: moment the resource occupied by train is released after completion of operation o_α;

b_α: index of the supply operation preceding operation o_α (operations o_{b_α} and o_α are executed by the same tugger train); $b_\alpha = 0$ means that o_α is the first operation in the system cycle;

f_α: index of the supply operation following o_α, (operations o_α and o_{f_α} are executed by the same tugger train).

Constraints:

I. transport process operations:

$$yt_\alpha = xt_\alpha + tr_\alpha, \quad \alpha = 1,2, \dots, \omega \,, \tag{1}$$

$$b_\alpha = 0, \forall \alpha \in BS, BS \subseteq BI = \{1,2, \dots, \omega\}, |BS| = l \tag{2}$$

$$b_\alpha \neq b_\beta \quad \forall \alpha, \beta \in BI \backslash BS, \ \alpha \neq \beta \,, \tag{3}$$

$$f_\alpha \neq f_\beta \quad \forall \alpha, \beta \in BI, \ \alpha \neq \beta \,, \tag{4}$$

$$(b_\alpha = \beta) \Rightarrow (f_\beta = \alpha), \ \forall b_\alpha \neq 0 \,, \tag{5}$$

$$[(b_\alpha = \beta) \wedge (b_\beta \neq 0)] \Rightarrow (yt_\beta + d_{\beta,\alpha} \leq xt_\alpha), \quad \alpha, \beta = 1,2, \dots, \omega \,, \tag{6}$$

$$xs_\alpha \geq yt_\alpha, \quad \alpha = 1,2, \dots, \omega \,, \tag{7}$$

$$[(f_\alpha = \beta) \wedge (b_\beta \neq 0)] \Rightarrow (xs_\alpha = xt_\beta - d_{\alpha,\beta}), \quad \alpha, \beta = 1,2, \dots, \omega \,, \tag{8}$$

$$[(f_\alpha = \beta) \wedge (b_\beta = 0)] \Rightarrow (yt_\alpha + d_{\alpha,\beta} \leq xt_\beta + \mathcal{TP}), \quad \forall o_\alpha, o_\beta \in \mathcal{P} \,, \tag{9}$$

$$[(f_\alpha = \beta) \wedge (b_\beta = 0)] \Rightarrow (yt_\alpha + d_{\alpha,\beta} \leq xt_\beta + \mathcal{TS}), \quad \forall o_\alpha, o_\beta \in \mathcal{S} \,, \tag{10}$$

$$[(f_\alpha = \beta) \wedge (b_\beta = 0)] \Rightarrow (xs_\alpha = xt_\beta - d_{\alpha,\beta} + \mathcal{TP}), \quad \forall o_\alpha, o_\beta \in \mathcal{P} \,, \tag{11}$$

$$[(f_\alpha = \beta) \wedge (b_\beta = 0)] \Rightarrow (xs_\alpha = xt_\beta - d_{\alpha,\beta} + \mathcal{TS}), \quad \forall o_\alpha, o_\beta \in \mathcal{S} \,, \tag{22}$$

II. transport operations and production requests

$$x_\alpha \leq xt_\alpha \leq x_\alpha + \Upsilon, \ \forall o_\alpha \in \mathcal{P}, . \tag{13}$$

$$x_\alpha - \Upsilon \leq yt_\alpha \leq x_\alpha, \ \forall o_\alpha \in \mathcal{S}, \tag{14}$$

III. collision and deadlock free

$$\left(\mathcal{K}_{\varepsilon\beta-\lambda\gamma}=1\right)\Rightarrow\left[\left(xt_{\beta}\le mod\{xs_{\lambda},TP(\lambda)\}\right)\vee\left(xt_{\gamma}\le mod\{xs_{\varepsilon},TP(\varepsilon)\}\right)\right] \tag{15}$$

$$\varepsilon,\lambda,\beta,\gamma=1,2,\ldots,\omega,$$

$$TP(\alpha)=\begin{cases}\mathcal{TP} \text{ if } o_{\alpha}\in\mathcal{P}\\ \mathcal{TS} \text{ if } o_{\alpha}\in\mathcal{S}\end{cases}.$$

Question:

Does there exist a set of routes Π operated by the given fleet TT, which ensures that a schedule xt_{α} follows all supply and pick-up requests x_{α}?

4.2 Method

The cyclic schedule of tugger trains TT_1, TT_2 routings (see Fig. 1) following streams of supply and pick-up requests while corresponding to the sequence of docking stations $\pi_1=(M_8,M_1,M_2,M_4,M_5,M_3)$ and $\pi_2=(M_9,M_6,M_7)$ respectively, is shown in Fig. 2. The supply requests following sequence π_1 are serviced with the takt time $\mathcal{TS}$ =600 u.t by tugger train TT_1, while pick-up requests following sequence π_2 are serviced with the takt time $\mathcal{TP}$ =800 u.t by tugger train TT_2. The assumed takt times determine the period of the milk-run system cyclic schedule equal to the least common multiple $NWW(\mathcal{TS},\mathcal{TP})=NWW(600,800)=2400$ u.t. During this period – hereinafter referred to as the time horizon H within the time window (see Fig. 2) – two tugger trains travelling the common shared critical region composed of transport section (G_9-G_8) are in a collision situation. This follows from an assumption supposing that a critical region consists of single carriageway sections not allowing them to be utilized by more than one tugger train at a time. Note that in the considered case tugger train TT_1 moves from M_4 to M_5 along (G_9-G_8) (highlighted by bold green line) while tugger TT_2 moves from M_9 to M_6 along sequence of sections: (G_{14}-G_{13}), (G_{13}-G_{10}), (G_{10}-G_8), (G_8- G_9),(G_9-G_7), (G_7-G_6) (highlighted by yellow line).

This observation leads to the conclusion that, since within the time horizon the tugger trains' collision may occur, the proper collision prevention or collision avoidance strategy has to be implemented. The strategy proposed hereafter follows from the above-provided observation showing that the tugger trains' collision occurs in the first periods of supply and pick-up request streams, respectively (see Fig. 2), more precisely in the takt time $\mathcal{TP}$. This observation suggests that, in case the collision avoidance can be guaranteed within $\mathcal{TP}$, a new profile of interacting processes can be considered. Consequently, processes' profile changing results in the same value of the least common multiple $NWW(\mathcal{TS},\mathcal{TP})$ imposing changes of operation times, however, without changing the frequency of events occurring in processes. Therefore, the all moments of request occurrences in an assumed time horizon can be subdivided into four calculation windows each one size of 600 u.t. (see Fig. 3). The solutions obtained for each calculation window while guaranteeing the collision avoidance of relevant tugger trains' routings results in collision-free milk-run traffic within the entire periodically repeating time horizon.

Due to this assumption, the considered Constraint Satisfaction Problem (CSP) essentially equates to the following recursive scheme:

$$CS_l=\left(\{CS_{(l-1)}\}\cup XT_l,\mathcal{D}_l,\ \mathcal{C}_l\right) \tag{16}$$

where:

l – number of calculation windows

$CS_1=(\{B,F,XT_1\},\mathcal{D}_1,\ \mathcal{C}_1)$ – CSP stated for the first calculation window; the set $\{B,F,XT_l\}$ states for its solution while following decision variables and satisfying the constraints $\mathcal{C}_1$.

$XT_l=\{xt_{\alpha}^{l}|\alpha=1,2,\ldots,\omega\}$ – set of moments xt_{α} for l-th calculation window

$\mathcal{D}_l$: a discrete finite set of domains of variables B,F,XT_l;

$\mathcal{C}_l$: a set of constraints describing the following relations in the l-th window: the order (1)–(8) and cyclic (9)-(12) execution of transport operations; the constraints which guarantee timely requests completion (13)-(14); the constraints ensure the collision- and deadlock-free execution of milk-run flows (15).

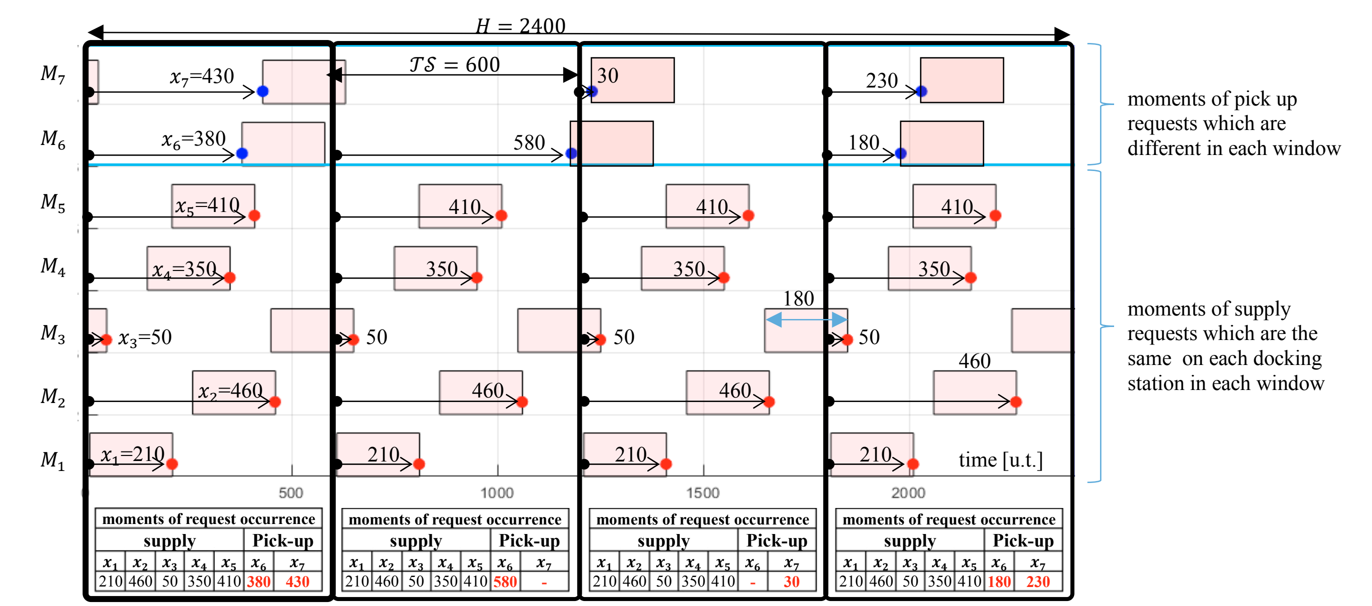

Fig. 3 The moments of requests occurrence in assumed time horizon subdivided into four calculation windows

To solve the CS_l problem (15), the sets of values of decision variables $\mathcal{V}_l$ (determining train routes B, F) and the set XT_l of moments xt_α from the l-th calculation window (i.e. delivery operation schedules), for which all the constraints $\mathcal{C}_l$ are satisfied, have to be deduced from preceding CS_{l-1}while taking into account different moments of requests x_α occurring in the l-th computational window.

In turn, to solve CS_{l-1} problem the sets $\mathcal{V}_{l-1}$ and XT_{l-1} of moments xt_α from the $l-1$-th calculation window, for which all the constraints $\mathcal{C}_{l-1}$ are satisfied, have to be deduced from the preceding CS_{l-2} while taking into account different moments of requests x_α occurring in the $l-1$-th computational window. And so on, up to $CS_2 = (\{CS_1\} \cup XT_2, \mathcal{D}_2, \mathcal{C}_2)$, where $CS_1 = (\{B, F, XT_1\}, \mathcal{D}_1, \mathcal{C}_1)$. In this context the CS problem integrates the issues of trains' routing (B, F) and scheduling of transport operations $(XT_1 \ldots XT_l)$.

These problems are usually handled using constraint programming environments, such as Oz Mozart, ILOG, and ECL^{i}PS^E (Bocewicz et al. 2019; Sitek and Wikarek 2017)

4.2 Computational results

Consider the milk-run system layout from Fig. 1. The goal is to find collision-free routings of a given tugger train's fleet (i.e. the set Π) following the streams of supply and pick-up requests (from Fig. 2).

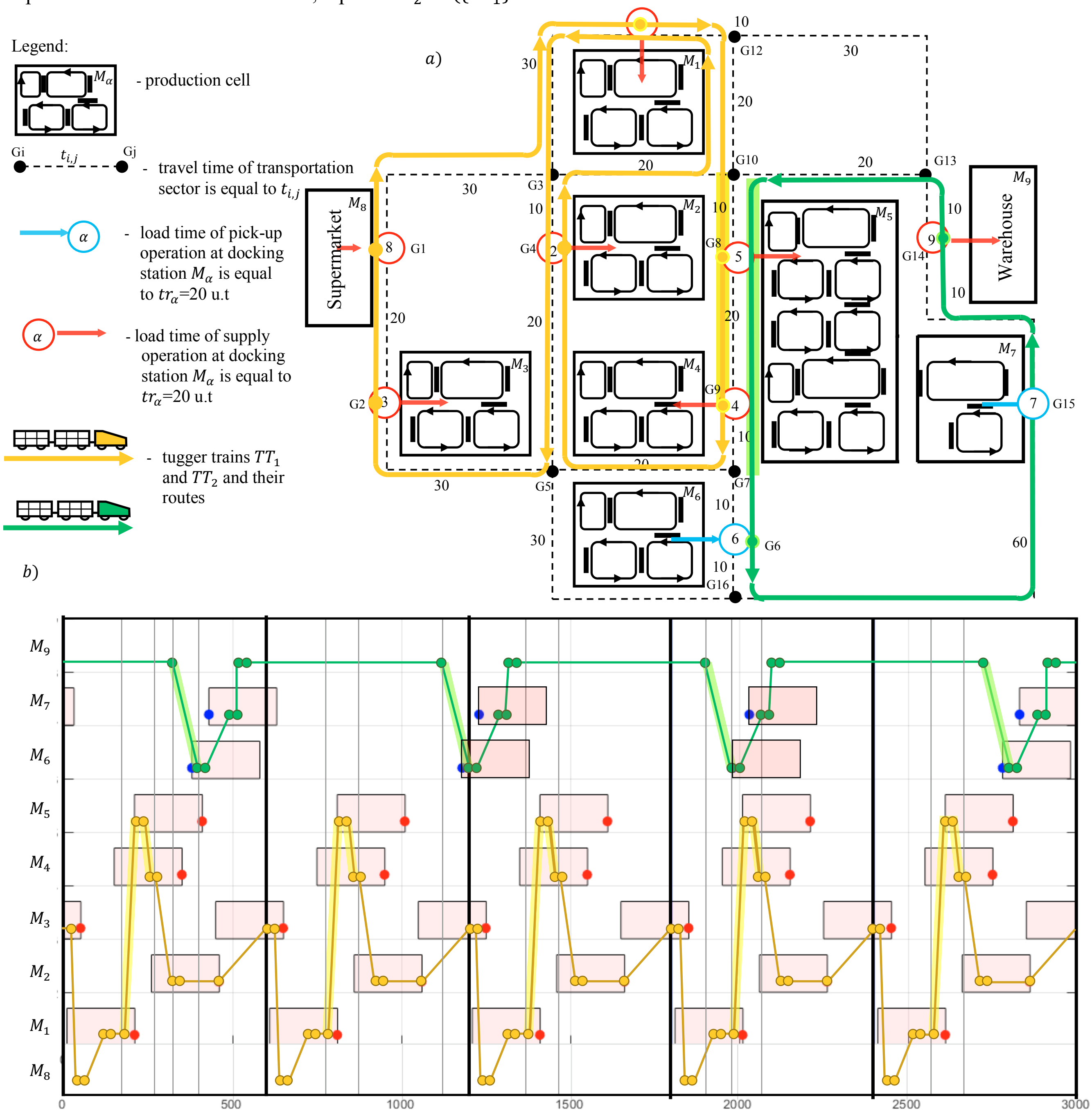

Fig. 4 The tugger trains TT_1,TT_2 routes a) and determined by them collision-free cyclic schedule b)

The solutions sought assume that:

- travel time $t_{i,j}$ by transportation sector $t_{i,j}$ is given and shown in Fig. 1
- travel time $d_{\alpha,\beta}$ between subsequent docking stations M_α and M_β is determined by the shortest path linking these stations,
- load time of pick-up and supply operations servicing docking station $M_\alpha$are the same and equal to tr_α=20 u.t.

To find the solution the problem CS_l (16) was solved for $l = 4$ calculation windows. The problem was implemented and solved in the constraint programming environment Oz Mozart (Windows 10, Intel Core Duo2 3.00 GHz, 4 GB RAM). The first admissible solution obtained in less than 10 seconds (shown in Fig. 5) provides final itinerary guaranteeing collision-free routings of tugger trains moving due to assigned cyclic schedule period which equals to the least common multiple of takt times $\mathcal{TS}$ and $\mathcal{TP}$.

Selected results of the experiments carried out for different numbers of docking stations (5-15) and different numbers of tugger trains (1-4) are provided in Table 1. The proposed declarative modelling driven approach is limited for the systems not exceed 15 docking stations.

Tab.1. Results of computational experiments for different number of docking stations (5-15) and different number of trains (1-4)

Numer of docking station	Number of tugger trains	Is there a solution?	Computation time [s]
5	1	✓	<1
5	2	✓	<1
5	3	✓	3
5	4	✓	5
9	1	✓	5
9*	2	✓	10
9	3	✓	20
9	4	✓	54
12	1	✓	35
12	2	✓	56
12	3	✓	124
12	4	✓	202
15	1	✓	184
15	2	✓	268
15	3	?	>900
15	4	?	>900

*the solution presented in the Fig. 4

5. CONCLUSIONS

The novelty presented in this study can be seen through an integrated modelling approach to milk-run system design and operation. In this context a complex approach proposed enables us to take into account both different, independently processed production flows, as well as them servicing different (independently executed of different profiles) pick-up and supply processes. Besides the modelling possibilities the recursive formulation of the constraint satisfaction problem provides an attractive tool allowing us to cope with the recursive design of cyclic schedules of delivery processes created by components seen as different profiles of pick-up and supply processes.

The results of the tests demonstrate that the proposed declarative framework standing behind the reference model of milk-run driven delivery and pick-up transportation problem formulated in terms of the recursive CSP model is a useful tool allowing us to handle services executed along periodic routings due to cyclic schedules. Implemented computationally, it enables fast online prototyping of supply schedules and transport routes of a tugger train fleet. The utilization of commercially available software tools, such as CPLEX/ECL^{i}PS^e/Gurobi, etc., allowing consideration of the scale of problems occurring in practice could, in turn, be seen as an attractive solution in a class of problem-oriented DSS.

Future research should be focused on finding sufficient conditions ensuring the prevention of milk-run flows collision while taking into account weaker constraints, for instance allowing us to consider routes linking subsequent docking stations other than determined by the shortest path. Aside from the research perspective presented in this paper, other directions of study worth mentioning are those aimed at investigating the conditions that would allow us to reschedule milk-run flows according to customers' changeable demands, i.e. production flows. Other potentially interesting areas of investigation for the future relate to the smooth transition between two successive cyclic steady states corresponding to the current and rescheduled flows.

REFERENCES

Bocewicz G., Nielsen P., Banaszak Z. (2019) Declarative Modeling of Milk-run Vehicle Routing Problem for Split and Merge Supply Streams Scheduling. Advances in Intelligent Systems and Computing . vol. 853, 157-172

Bocewicz G., Banaszak Z., Nielsen I., (2017) Delivery-flow routing and scheduling subject to constraints imposed by vehicle flows in fractal-like networks, Archives of Control Sciences, Vol. 27(LXIII), No. 2, 135–150

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

Gyulaia D., Pfeiffer A., Sobottka T., Váncza J.(2013) Milkrun Vehicle Routing Approach for Shop-floor Logistics, Forty Sixth CIRP Conference on Manufacturing Systems 2013, Procedia CIRP 7, 127 – 132, doi: 10.1016/j.procir.2013.05.022

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

Lenstra J.K., Rinnooy Kan A.H.G (1981), Complexity of Vehicle and Scheduling, Problems, Networks 11, 221-227

Nguyen P.K., Crainic T.G., Toulouse M., (2017) Multi-trip pickup and delivery problem with time windows and synchronization, Annals of Operations Research, Springer, Vol. 253 (2), 899-934

Ong J.O., Suprayogi (2011) Vehicle Routing Problem with Backhaul, Multiple Trips and Time Window, Jurnal Teknik Industri, Vol. 13, No. 1, ISSN 1411-2485 print / ISSN 2087-7439 online

Patel D., Patel M. B., Vadher J. A. (2014) Implementation of milk run material supply system in vehicle routing problem with simultaneous pickup and delivery, International Journal of Application or Innovation in Engineering & Management (IJAIEM), Vol. 3, Issue 11, 2014, 122-124, ISSN 2319 – 4847

Perronnet F., Abbas-Turki A., El Moudni A., (2014) Vehicle routing through deadlock-free policy for cooperative traffic control in a network of intersections: Reservation and congestion, 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, 2014, 2233-2238. doi: 10.1109/ITSC.2014.6958034

Pillac V. Gendreau M., Guéret C., Medaglia A.L. (2013) A Review of Dynamic Vehicle Routing Problems, European Journal of Operational Research, 225(1), 1-11

Polak M., Majdzik P., Banaszak Z., Wójcik R. (2004) The performance evaluation tool for automated prototyping of concurrent cyclic processes. Fundamenta Informaticae. Vol. 60, No.1-4, 269-289

Sitek P., Wikarek J. (2017) Capacitated Vehicle Routing Problem with Pick-up and Alternative Delivery (CVRPPAD) – model and implementation using hybrid approach, Annals of Operations Research

Suprayogi, Priyandari Y. (2009) Vehicle Routing Problem with Multiple Trips, Time Windows, and Simultaneous Delivery and Pickup Services. Asia Pacific Industrial Engineering and Management System. Vol. 8, 1543-1552.

Wysk R.A., Yang N.-S., Joshi S., (1994) Resolution of deadlocks in flexible manufacturing systems: Avoidance and recovery approaches, Journal of Manufacturing Systems, Vol. 13, Issue 2, 128-138