

Collective decision making in dynamic vehicle routing problem

Edyta Kucharska^{1*}, Katarzyna Grobler-Dębska¹, and Radosław Klimek²

¹AGH University of Science and Technology, Department of Automatics and Robotics, 30 Mickiewicza Av, 30-059 Krakow, Poland

²AGH University of Science and Technology, Department of Computer Science, 30 Mickiewicza Av, 30-059 Krakow, Poland

Abstract. The paper presents a collective decision making in dynamic vehicle routing problem. In contrast to the static problem, a part or all of the customers' companies are revealed dynamically during the design or execution of the routes. The problem is modelled using the algebraic-logical meta-model (ALMM) methodology, which enables making collective decisions in successive process stages, not separately for individual vehicles. ALMM is considered to be the rule according to which the availability of companies is determined. The steps and schematics of the general algorithm that take into account the dynamic appearance of new companies are shown. The proposed approach belongs to trajectory-based metaheuristics methods.

1 Introduction

The aim of the paper is to present ALMM-based approach for collective decision making in dynamic vehicle routing problem. In dynamic problems, a part or all of the input is revealed dynamically during the design or implementation of a statically planned solution. An invention of new ways to work in real factories (*e.g.* new complex production lines and automation of production) and logistic systems (*e.g.* mobile phones and global positioning systems) creates the need for new solutions and the development of existing methods. Moreover, a holistic approach is becoming increasingly popular, rather than those focusing on individual executors and other resources.

The proposed approach is based on the algebraic-logical meta-model (ALMM) [4],[7], which enables making collective decisions in successive process stages, not separately for individual objects or executors [8],[14]. This approach may take into account the basic and specific limitations of the problem (*e.g.* deadline, resources dependent on the process state). Moreover, the ALMM approach allows reconstructing the process of decision making as well as monitoring and tracking decisions during the manufacturing or the logistic process. It should be emphasised that the decision is made jointly and takes into account the current situation (current process state). Modelling and simulation of collective decision-making in the logistic process belong to artificial intelligence method.

2 Motivation

The real-world scheduling problems are one of the most challenging because most of them are known to be NP-complete or NP-hard in their general and also dynamic forms (*i.e.* problem data are changing while the optimisation is in progress). For example, in the

Dynamic Vehicle Routing Problem (DVRP), the aim consists in designing the optimal set of routes for a fleet of vehicles in order to serve a given set of customers. However, new customer orders arrive while the working day plan is in progress. In this case, routes must be reconfigured dynamically while executing the current simulation [12]. Dynamic problems are much harder to solve than combinatorial optimisation problems, which are usually static and all data are known in advance, *i.e.* before the optimisation process has started. For that reason, dynamic VRP is under consideration of many research studies. Moreover, dynamic problems should be considered in the context of increasingly popular modern urban systems and smart cities [13].

The paper [19] is a survey that classifies routing problems from the perspective of information quality and evolution. A number of variants have been studied: Capacitated VRP (CVRP), where each customer has a demand for a good and vehicles have finite capacity; the VRP with Time Windows (VRPTW), where each customer must be visited during a specific time frame; the VRP with Pick-up and Delivery (PDP), where goods have to be picked up and delivered in specific amounts at the vertices; and the Heterogeneous fleet VRP (HVRP), where vehicles have different capacities. Routing problems that involve moving people between locations are referred to as Dial-A-Ride-Problem (DARP) for land transport; or Dial-A-Flight-Problem (DAFP), for air transport.

In the paper [12] the authors present the dynamic VRP, its interests in practical applications, and its specific characteristics and also a review of common solutions for the representation of the problem and the major classes of metaheuristics proposed to solve this problem. The authors present two major classes of metaheuristics: Trajectory-Based and Populations-Based. They assigned to the first group of classes Tabu Search

* Corresponding author: edyta@agh.edu.pl

[21], Greedy Randomized Adaptive Search Procedure (GRASP) and variable neighbourhood search (VNS) [1], [20]. To the second group belong Ant Colony [18], Evolutionary Algorithms and Particle Swarm Optimization. It should be emphasised that all these metaheuristics have been appropriately adapted to the types of dynamic variables in the considered problems or a combination of two or more metaheuristic [10], [16]. Also approaches based on dynamic programming could be classified as trajectory based simulations [3], [17]. In the literature, other classes of approaches to dynamic VPR are also considered, such as Scenario Based Planning [2] and Logic and Graph Theory-based approach [11], [22].

The approach presented in this paper is the trajectory-based simulation. Previously, the supply routes for multi-location companies problem was considered [9], it is a special case of the considered problem. This paper presents an approach based on the ALMM methodology, which takes into account the possibility of modelling the dynamic appearance of new locations to visit during the simulation by adding an appropriate general rule.

3 Problem Description

Let us consider a supplier company that provides goods to several customers' companies and the goods delivery is carried out by the company cars. The supplier employs a number of sales representatives (vehicles) to sell and deliver goods to customers' offices. A dynamic problem is considered, *i.e.* a part or all of the customers' companies are revealed dynamically during the design or execution of the routes. There may be different reasons for this. In the simplest case, the disclosure times are given in advance. In another case, the availability of companies depends on the availability of other companies or other conditions resulting from arrangements between companies. For example, in the case of multi-location companies, there is a need to visit the head office first (in order to negotiate, to establish the transaction conditions, *etc.*) and then delivery to their branches is possible. The article considers the general rule, according to which it can be determined from when it is possible to deliver to given company.

The problem under consideration can be stated as follows. The supplier company is assumed, in which there are given: a team of supplier travelling vehicles and a set of locations to visit. Each location represents a customer company (office-head or branches are not distinguished in the case of a multi-location company). There is also one initial location for vehicles (*i.e.* supplier company office). Each location (except the initial) must be visited exactly once, and any vehicle from the team can do it. Vehicles start from the initial location and return to it in the end. A location can be visited only if conditions resulting from arrangements between companies are met (they define the rules for making the company available to visit). The distances between locations are known. The aim of the optimisation is to plan vehicle routes so that the whole set of locations is visited in the shortest time.

The following notation is used: $M = \{m_1, m_2, \dots, m_{|M|}\}$ is a set of vehicles and $N = \{n_0, n_1, n_2, \dots, n_{|N|}\}$ is a set of locations, where the initial location is denoted as n_0 and n_i for $i = 1, \dots, |N|$ denotes locations to visit. In addition, let a_i denote the available location to visit and a set $A \subset N \setminus \{n_0\}$ is a set of all available location to visit. It means that the number of A set elements is changing (is increasing). The available locations to visit in the initial state are known. The dates of availability of subsequent locations are calculated according to a predetermined rule $r(s)$, which means that the individual's availability dates of locations to visit are dependent on the state of the system and for the n_j location is denoted as $r_{n_j}(s)$.

The distances between locations are presented in the distance matrix D, where individual elements d_{ij} are the distances between the location n_i and the location n_j , where $i, j = 0, 1, 2, \dots, |N|$ and $i \neq j$. All d_{ii} elements are infinite. Two kinds of vehicles are considered, a car or a truck. A vehicle travels with the speed v , where a car travels with the v_1 speed and a truck travels with a speed v_2 , because of traffic regulations and technical capabilities. In addition, the time spent at every location is omitted. The aim of optimisation is makespan C_{max} . Thus, dynamic vehicle routing problem with a heterogeneous fleet (D-HVRP) is examined.

Note that the well-known vehicle routing problem is NP-hard, thus, the considered problem also belongs to the class of NP-hard problems.

4 Algebraic-logical Model of Problem

The described problem is modelled as a multistage decision process.

4.1 ALMM definition

Let us recall ALMM formal definition, devised by Dudek- Dyduch [4], [7].

Algebraic-logical meta-model of multistage decision process is a process that is defined by the sextuple $MDP = (U, S, s_0, f, S_N, S_G)$ where: U is a set of decisions, $S = X \times T$ is a set of generalised states (X is a set of proper states and $T \subset \mathbb{R}^+ \cup \{0\}$ is a subset of non-negative real numbers representing the time instants), $f: U \times S \rightarrow S$ is a partial function called a transition function (it does not have to be defined for all elements of the set $U \times S$), $s_0 = (x_0, t_0)$ is an initial generalised state, $S_N \subset S$ is a set of not admissible generalised states, $S_G \subset S$ is a set of goal generalised states, *i.e.* the states in which we want the process to be at the end.

Transition function f is defined by means of two functions, *i.e.* $f = (f_x, f_t)$ where: $f_x: U \times X \times T \rightarrow X$ determines the next state and $f_t: U \times X \times T \rightarrow T$ determines the next time instant. As a result of the decision u that is taken at some proper state x and a moment t , the state of the process changes to $x' = f_x(u, x, t)$ that is observed at the moment $t' = f_t(u, x, t) = t + \Delta t$.

Since not all decisions defined formally make sense in certain situations, the transition function f is defined as a partial one. Consequently, all limitations concerning

the control decisions in a given state s can be defined in a convenient way by means of so-called sets of possible decisions $U_p(s) = \{u \in U: (u, s) \in \text{Dom } f\}$.

The values of particular co-ordinates of a state and decision do not have to be only numerical, but may be the names of elements (symbols) as well as certain objects (e.g. a finite set, sequence *etc.*).

To define a particular optimisation problem in ALMM methodology, one should build an algebraic-logical model of the problem P and give a specified optimisation criterion Q . The optimisation task is to find an admissible decision sequence \tilde{u} that optimises criterion Q . Thus, the optimisation problem is defined by the pair (P, Q) .

4.2 Algebraic-logical model of the problem

In this section an algebraic-logical model of the problem in question is described: the state of the system, the set of non-admissible states and the set of goal states, the decision, set of possible decisions and the transition function.

4.2.1 Process state

The process state $s = (x, t)$ in particular moment t can be described by the current state of all vehicles and a set of visited locations. The proper state x is defined as:

$$x = (x^0, x^1, \dots, x^{|M|}, x^{|M|+1}) \quad (1)$$

where:

- x^0 - set of locations, that have been visited until moment t ,
- x^k - state of k -th vehicle, for $k = 1, 2, \dots, |M|$,
- $x^{|M|+1}$ - set of locations available at the current moment t .

At any given time a vehicle can either go to some location, or remain at the last visited location. The state of a k -th vehicle at a time t is as follows: $x^k = (\eta, \rho)$, where values of individual variables are:

- $\eta \in N \cup n_0$ - location the vehicle is going to or location at which the vehicle stays,
- $\rho \in \mathfrak{R}^+ \cup \{0\}$ - length of road which the vehicle has to travel to reach the assigned location. If the vehicle is staying at some location, the value of ρ is equal to 0.

The initial generalised state $s_0 = (x_0, t_0)$ of the process is as follows:

$$x_0 = (x_0^0, x_0^1, \dots, x_0^{|M|}, x_0^{|M|+1}) \quad (2)$$

In the initial state (when $t_0 = 0$), all vehicles are at the initial location n_0 , therefore $x_0^k = (n_0, 0)$. There is no location visited at the time t_0 , so the set x_0^0 is empty: $x_0^0 = \emptyset$. Not all locations to visit are available in the initial state, thus the set of locations available at the time t_0

$$x_0^{|M|+1} = A.$$

It is said that k -th vehicle is idle (not working) in state s when it is staying at the location n_i and the next location can be assigned to visit. Therefore the state of k -th vehicle: $x^k(s) = (n_i, 0)$. A vehicle is working (is busy) in a given state s when it is going to a designated

location or has finished the journey and returned to the initial location n_0 :

$$x^k(s) \in \{(\eta, \rho): \eta = n_i, i \neq 0, \rho > 0 \vee \eta = n_0, t(s) > 0\} \quad (3)$$

Set of non-admissible generalised states S_N includes states in which all vehicles have returned to the initial location and not all locations have been visited:

$$S_N = \{s = (x, t): x^0(s) \neq N \wedge \forall_{k=1, \dots, |M|}; (x^k(s) = (n_0, 0)) \wedge t > 0\} \quad (4)$$

Set of goal states S_G includes states in which all the locations have been visited and all vehicles have returned to the initial location n_0 . Therefore, the set of goal states S_G is as follows:

$$S_G = \{s = (x, t): s \notin S_N \wedge x^0(s) = N \wedge \forall_{k=1, \dots, |M|}; (x^k(s) = (n_0, 0)) \wedge x^{|M|+1}(s) = N\} \quad (5)$$

4.2.2 Decisions

In a given state s a decision must be taken. It is not determined separately for one particular vehicle but collectively for all vehicles. For that reason, the decision is defined as a vector $u \in U = U^1 \times U^2 \times \dots \times U^{|M|}$, instead of separate values for particular vehicles. The particular coordinate $u^k \in U^k$ represents separate decisions and refers to the k -th vehicle ($k = 1, \dots, |M|$).

It should be emphasised that because the problem is dynamical the values of the possible decision are different at different states; that is why a set of decisions from the state of $u(s)$ is considered.

Therefore, during the collective decision making, the joint operation of all vehicles, not the sum of the activities of individual vehicles is assessed. The decision chosen is the best because of the common goal. In addition, not all decisions are possible at every stage due to the existing constraints. The decision is picked from the set of possible decisions $U_p(x, t)$, *i.e.* decisions that can be taken in the particular state s . In the problem under consideration, the decision is to determine to which locations particular vehicles should go at the moment t . The location to visit must be available at this state. The time of location availability is not known in advance because it depends on the state of the whole system. Therefore, a rule $r(s)$, checking whether the set of available locations should be increased by a new available location, should be calculated in each state.

It was assumed that the decision assigning a vehicle the next location to visit can be taken only after it reaches the previously assigned location. The decision cannot be changed for a vehicle which has not yet reached its assigned location (a decision can be only to continue the travel). For a vehicle staying at a certain location, a decision can be to visit another location, to return to the initial location or remain at the current location.

Thus, the decision in the state s is a vector, $u(s) = (u^1(s), \dots, u^{|M|}(s))$ and the value of particular co-ordinate $u^k(s) \in x^{|M|+1}(s)$ is as follows:

- if the k -th vehicle is working and n_i location is assigned to the k -th vehicle (*i.e.* $x^k = (n_i, \rho)$, where $\rho > 0$), then only the possible decision to take is traveling continuation $u^k = n_i$,
- if the k -th vehicle is idle (is staying at n_i location, *i.e.* $x^k = (n_i, 0)$) and there is no location that could be assigned ($x^{|M|+1} = \emptyset$) but not all locations have been visited or is assigning to visit *i.e.* ($N \neq x^0(s) \cup \{n_i: \exists k = 1, \dots, |M|: x^k(s) = (n_i, \rho), \rho > 0\}$), then the only possible decision to take is not assigning any new location, the k -th vehicle is still staying at n_i location, $u^k = n_i$,
- if the k -th vehicle is idle (is staying at n_i location, *i.e.* $x^k = (n_i, 0)$) and there is an available location that could be assigned ($x^{|M|+1} \neq \emptyset \wedge n_j \in x^{|M|+1}$), then the possible decision to take is assigning any new location n_j ($u^k = n_j$),
- if the k -th vehicle is idle is staying at n_i location, *i.e.* $x^k = (n_i, 0)$ and there is no location that could be assigned ($x^{|M|+1} = \emptyset$) and all locations have been visited or is assigning to visit, *i.e.* $N = x^0(s) \cup \{n_i: \exists k = 1, \dots, |M|: x^k(s) = (n_i, \rho), \rho > 0\}$, then only possible decision to take is assigning the initial location n_0 , $u^k = n_0$.

Let us denote the set of assigned locations in a given state s as $\overline{N}(s)$:

$$\overline{N}(s) = \{n_i: \exists k = 1, \dots, |M|: x^k(s) = (n_i, \rho), \rho > 0\} \quad (6)$$

The decision u_p must belong to the set of possible decision U_p in a given state s . A possible decision is dependent on the state of the particular vehicle. Thus, possible decisions $u_p^i(s)$ for the i -th vehicle in state $s = (x, t)$ are defined as follows:

$$u_p^i(s) = \begin{cases} n_i & \text{for } \begin{aligned} &x^k(s) = (n_i, \rho) \vee \\ &\vee (x^k(s) = (n_i, 0) \wedge x^{|M|+1}(s) = \emptyset) \end{aligned} \\ n_j & \text{for } \begin{aligned} &x^k(s) = (n_i, 0) \wedge x^{|M|+1}(s) \neq \emptyset \wedge \\ &\wedge n_j \in x^{|M|+1}(s) \wedge N \neq x^0(s) \cup \overline{N}(s) \end{aligned} \\ n_0 & \text{for } x^k(s) = (n_i, 0) \wedge N = x^0(s) \cup \overline{N}(s) \end{cases} \quad (7)$$

The complete definition of the set of the possible decision is as follows:

$$U_p(s) = U_p^1(s) \times \dots \times U_p^{|M|}(s) \setminus H(s) \quad (8)$$

where $H(s)$ is a set of decisions assigning the same location to more than one vehicle at the same time except that all locations have been visited and the vehicle return to the initial location n_0 :

$$H(s) = \{u(s): \exists k=1, \dots, |M| \exists l=1, \dots, |M|: k \neq l \wedge u^k(s) = u^l(s) = n_i \wedge n_i \neq n_0 \wedge N \neq x^0(s) \cup \overline{N}(s)\} \quad (9)$$

4.2.3 Transition function

Based on the current state $s = (x, t)$ and the decision taken in this state $u(s)$, the subsequent state $(x', t') =$

$f(u, x, t)$ is generated by means of the transition function f . The transition function is defined for each possible decision $u(s) \in U_p(s)$ and consists of two stages.

First, it is necessary to determine the moment t' when the subsequent state occurs. It is the nearest moment in which at least one vehicle has reached the previously assigned location or the next moment the location is available. The subsequent state x' will occur at the moment $t' = t + \Delta t$, where Δt equals the lowest value of the established completion times:

- if k -th vehicle, staying at location n_i , is idle and has an assigned location n_j to visit, the completion time is equal to $t^k = \frac{d_{ij}}{v}$, where $v = v_1$ for car or $v = v_2$ for truck,
- if k -th vehicle is going to a previously assigned location n_i and the length of the remaining part of the road which vehicle has to travel to reach the assigned location is equal to ρ (*i.e.* $x^k = (n_i, \rho), 0 < \rho < d_{ij}$), then the completion time is equal to $t^k = \frac{\rho}{v}$, where $v = v_1$ for car or $v = v_2$ for truck,
- if k -th vehicle is idle and no new location is assigned, the completion time is equal to the minimum time at which a new location will be available:

$$t^k = \min \left\{ \tau_{n_j}(s): n_j \in N \setminus \{x^0(s) \cup \overline{N}(s) \cup n_0\} \right\} \quad (10)$$

Once the moment t' is known, it is possible to determine the proper state of the process at that time.

The set of visited locations

The set of visited locations (first proper state coordinate) x^0 , is modified by adding each location reached by one of the vehicle at time t' :

$$x^0(s)' = x^0(s) \cup \{n_i \in N: \exists k=1, 2, \dots, |M| u^k(s) = n_i \wedge t^k = \Delta t\} \quad (11)$$

The state of vehicle

The values of subsequent coordinates x^k for $k = 1, 2, \dots, |M|$ represent the states of vehicles. The state $x^k = (\eta, \rho)$ goes into $x^{k'} = (\eta', \rho')$, for $k = 1, 2, \dots, |M|$ and is determined as follows:

- if the k -th vehicle is going to location n_i (*i.e.* vehicle is working - $x^k = (n_i, \rho)$, where $\rho > 0$) and decision is continuation of travel $u^k = n_i$:

$$\begin{aligned} \eta' &= n_i \\ \rho' &= \begin{cases} 0 & \text{for } t^k = \Delta t \\ \rho - v * \Delta t & \text{for } t^k > \Delta t \end{cases} \end{aligned} \quad (12)$$

- if the k -th vehicle is staying at n_i location (*i.e.* the vehicle is idle - $x^k = (n_i, 0)$) and there is no location that could be assigned ($x^{|M|+1} = \emptyset$) but not all locations have been visited or is assigning to visit *i.e.* ($N \neq x^0(s) \cup \overline{N}(s)$) and decision is still staying at n_i location, $u^k = n_i$:

$$\begin{aligned} \eta' &= n_i \\ \rho' &= 0 \end{aligned} \quad (13)$$

- if the k -th vehicle is staying at n_i location, (i.e. $x^k = (n_i, 0)$) and there is available location that could be assigned ($x^{|M|+1} \neq \emptyset \wedge n_j \in x^{|M|+1}$) and decision is assigning any new location n_j ($u^k = n_j$):

$$\begin{aligned} \eta' &= n_j \\ \rho' &= d_{ij} \end{aligned} \quad (14)$$

- if the k -th vehicle is staying at n_i location, (i.e. $x^k = (n_i, 0)$) and there is no location that could be assigned ($x^{|M|+1} = \emptyset$) and all locations have been visited or is assigning to visit (i.e. $N = x^0(s) \cup \overline{N}(s)$) and decision is assigning the initial location n_0 , $u^k = n_0$:

$$\begin{aligned} \eta' &= n_0 \\ \rho' &= d_{i0} \end{aligned} \quad (15)$$

The set of available locations

The set of locations available in a given state (coordinate $x^{|M|+1}$) is modified by reduction of the locations which have been just visited or assigned to visit by vehicle and adding each location, whose availability time, calculated from rule $r(s)$ is in time window between current state and next state and the initial state n_0 is always available:

$$\begin{aligned} x^{|M|+1}(s)' &= x^{|M|+1}(s) \setminus \{x^0(s) \cup \overline{N}(s)\} \cup n_0 \cup \\ &\cup \{n_j; r_{n_j}(s) \in (t, t + \Delta t)\} \end{aligned} \quad (16)$$

5 Algorithm

The algorithms based on the algebraic-logic model consist of consecutive construction system trajectory (or trajectories) [5-9],[14],[15].

The general corresponding algorithm for generating a single trajectory for the problem under consideration is as follows. The trajectory is generated from the initial state $s_0 = (x_0, t_0)$. Each next state $s' = (x', t')$ depends on the previous state $s = (x, t)$ and the decision $u(s)$ made at that state. The decision $u(s)$ is chosen from a set of possible decisions at the given state, using a choosing technique. Then, for the given state and chosen decision a new process state $s' = (x', t')$ is determined. The process of the proper state and the corresponding moment of time are calculated using the transition function of the process $f(x, t, u)$. Firstly, moment t' is determined as the minimum value from the time the vehicles complete their activity and the time of availability of the new locations, calculated by the rule $r(s)$. Secondly, the proper state of the process x' is determined i.e. the set of visited locations, the state of a particular vehicle and the set of available locations. Then it is checked whether the new system state belongs to the set of non-admissible generalised states; if so, the trajectory generation is stopped (the trajectory is non-admissible). The next step is to check the membership of the goal states set. If the new state belongs to this set, the generation of the trajectory is completed successfully and the identification of the solution can be made. Figure 1 shows the schema of the algorithm.

A very important element of the algorithm is the technique for choosing decision $u(s)$. The decisions can be determined by local decision-making procedures. The

literature presents different heuristic methods of discrete optimisation problems using local optimisation, which are based on minimisation (maximisation) of the local increase of quality criterion. Moreover, based on the properties of the ALMM methodology, it is able to create sophisticated local optimisation criteria, which take into account much more information than merely the increase of the criterion (for example unfavourable states or distinguished favourable states). Another way to determine the decision in a given process state is by using specially constructed rules, which determine only one or a few (usually not all) of the decisions belonging to the set of possible decisions $U_p(s)$. A specific algorithm requires establishing the decision-making technique. Based on the ALMM methodology a special type of metaheuristics and algorithms are developed ([6-9], [14]) and can be used to determine local decision-making procedures in this specific problems.

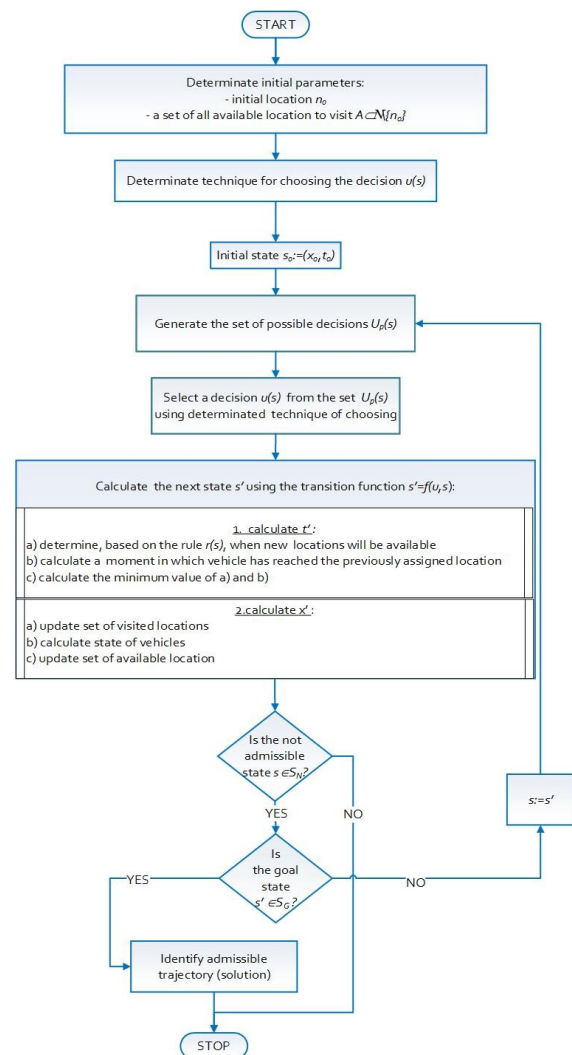


Fig. 1. Schema of algorithm.

6 Conclusion

The paper presents the approach based on the ALMM methodology, which takes into account the possibility of modelling the dynamic appearance of new locations to visit, during the simulation, by adding an appropriate

general rule. It has been shown that this approach enables defining flexible rules to model a dynamic process. The future research plan to add predictive methods to support the decision-making process with additional knowledge to the model.

This research is supported by AGH University of Science and Technology, Contract No. 11.11.120.417 and 15.11.120.625.

References

1. N. Azi, M. Gendreau, J. Y. Potvin, *A dynamic vehicle routing problem with multiple delivery routes* Annals of Operations Research, 199(1), 103-112 (2012)
2. R. W. Bent, P. Van Hentenryck, *Scenario-based planning for partially dynamic vehicle routing with stochastic customers* Operations Research, 52(6), 977-987 (2004)
3. M. Caramia, G. Italiano, G. Oriolo A. Pacifici, A. Perugia, *Routing a fleet of vehicles for dynamic combined pick-up and deliveries services* Operations Research Proceedings 2001 (pp. 3-8). Springer, Berlin, Heidelberg (2002)
4. E. Dudek-Dyduch, *Formalization and Analysis of Problems of Discrete Manufacturing Processes*. Scientific Bulletin of AGH University, Automatics Volume 54, Krakow, Poland (1990); (In Polish)
5. E. Dudek-Dyduch, *Learning based algorithm in scheduling*. J. Intell. Manuf., 11, 135-143 (2000)
6. E. Dudek-Dyduch, T. Dyduch, *Learning algorithms for scheduling using knowledge based model* Artif. Intell. Soft Comput.–ICAISC, 1091-1100 (2006)
7. E. Dudek-Dyduch, *Algebraic Logical Meta-Model of Decision Processes—New Metaheuristics* In proceedings of International Conference on Artificial Intelligence and Soft Computing, Zakopane, Poland, 14-18 June 2015, Lecture Notes in Computer Science, Springer Verlag, vol. 9119, 541-554 (2015)
8. E. Dudek-Dyduch, E. Kucharska, *Learning Method for Co-operation*. In proceedings of International Conference on Computational Collective Intelligence, Gdynia, Poland 21-23 September 2011. , Lecture Notes in Computer Science, Springer-Verlag, 6923, 290-300 (2011)
9. L. Dutkiewicz, E. Kucharska, K. Rączka, Grobler-K. Dębska, *ST method-based algorithm for the supply routes for multilocation companies problem* In Knowledge, Information and Creativity Support Systems: Recent Trends, Advances and Solutions. Springer, Cham. 123-135 (2016)
10. J. Euchi, A. Yassine, H. Chabchoub, *The dynamic vehicle routing problem: Solution with hybrid metaheuristic approach*, Swarm and Evolutionary Computation, 41 -53, (2015)
11. K. Jobczyk, P. Wiśniewski, A. Ligeza *Temporal Traveling Salesman Problem—in a Logic-and Graph Theory-Based Depiction* In International Conference on Artificial Intelligence and Soft Computing, Springer, Cham. 544-556 (2018)
12. M. R.Khouadjia, B. Sarasola, El-G. Talbi, L. Jourdan, *Metaheuristics for Dynamic Vehicle Routing* In E. Alba, A. Nakib, P. Siarry ed. Metaheuristics for Dynamic Optimization, Springer Berlin Heidelberg", 265-289 (2013)
13. R. Klimek, L. Kotulski, *Towards a better understanding and behavior recognition of inhabitants in smart cities. A public transport case* In proceedings of International Conference on Artificial Intelligence and Soft Computing, Zakopane, Poland, 14-18 June 2015, Lecture Notes in Computer Science, Springer Verlag, vol. 9120, 237-246 (2015)
14. E. Kucharska, *Heuristic Method for Decision-Making in Common Scheduling Problems* Applied Sciences, 7(10), 1073 (2017).
15. E. Kucharska, K. Grobler-Dębska, K. Rączka, *Algebraic-logical meta-model based approach for scheduling manufacturing problem with defects removal* Advances in Mechanical Engineering, 9(4), 1687814017692291. (2017)
16. R.J. Kuo, B.S. Wibowo, F.E. Zulvia *Application of a fuzzy ant colony system to solve the dynamic vehicle routing problem with uncertain service time*, Applied Mathematical Modelling, 9990 - 10001, (2016).
17. C. Novoa, R. Storer, *An approximate dynamic programming approach for the vehicle routing problem with stochastic demands* European Journal of Operational Research, 196(2), 509-515 (2009)
18. R. Montemanni, L. M. Gambardella, A. E. Rizzoli, V. V. Donati, A. *Ant colony system for a dynamic vehicle routing problem* Journal of Combinatorial Optimization, 10(4), 327-343 (2005)
19. V. Pillac, M. Gendreau, C. Guéret, A. L. Medaglia, *A review of dynamic vehicle routing problems* European Journal of Operational Research, 225(1), 1-11 (2013)
20. J. Yang, P. Jaillet, H. Mahmassani, *Real-time multivehicle truckload pickup and delivery problems* Transportation Science, 38(2), 135-148 (2004)
21. X. Yangkun, F. Zhuo, *Improved tabu search algorithm for the open vehicle routing problem with soft time windows and satisfaction rate*, Cluster Computing, (2018)
22. P. Wiśniewski, K. Kluza and A. Ligeza, *Decision support system for robust urban transport management*, Federated Conference on Computer Science and Information Systems (FedCSIS), Prague, pp. 1069-1074. (2017)