Masters in Computer Science

Dial-A-Ride Problem (DARP) Simulation

Kushagra Singh Bisen
M1 - Cyber Physical Social Systems

17th December 2020.
Dernière modification : 4 janvier 2021

I. Dial-A-Ride Simulation.

A. Problem Description.

The **Dial-A-Ride problem** (DARP) generalizes a number of well studied vehicle routing problems such as the Vehicle Routing Problem with Time Window (VRPTW) in Toth and Vigo (2001) and the Pickup and Delivery Problem (PDP) in Parragh et al (2008). The DARP is a demand responsive transportation service, which arises for example in the context of patient transportation.
This is a dynamic resource allocation problem of traveler requests (resources) to vehicles (consumers). The strategy to solve this problem is performed by agents, i.e the vehicles, which have to process at least 90% of the traveler requests. Agents are mobile, distributed. By hypothesis, at beginning the requests are randomly created and appears everywhere in the environment. Each request has a lifespan that is also randomly generated. A request is satisfied if a vehicle is at its location to serve it during the period validity of the request. The content of the request is the destination of the traveller. When a vehicle serves a request, it computes its new route to drive to the traveller destination. At destination, the vehicle is released. The strategy objective is to maximize the number of satisfied requests with the minimum of vehicles. Your strategy will be compared to a random strategy (vehicle moves randomly and serves requests when it is possible (at location and free.))

Formal Multi Agent Model

Based on the AEIO approach, we proposed formal multiagent model for this problem with a MAS based on cognitive agents (DARPC).
For simplification purpose,

1. Vehicles move in an Euclidian space.
2. Vehicle cannot satisfy simultaneously several requests (i.e several client like in carpooling.)

At minima (you can propose other questions), answer to these questions for DARPR and DARPC model.

1. Agents : what are agents and their behaviour ? => basic behaviour and decision process.
2. Environment : what are its properties ? => what belongs to the environment ? what is perceptible ? what is modifiable ?
3. Interaction : How agents influence behaviours of the other ? => which information is useful and how / when the information is exchanged> There is indirect indirection ?
4. Organization : What rules the behaviour of agents ? => Agents have permissions, obligations ?

Repast Operational Model

The objective of this project based course is to design a Repast based Simulation for DARPC with the following communication hypothesis, each vehicle agent can communicate with all others. The "REPAST with Java - Getting Started" is the tutorial used to learn the simulation and a basic version of the DARPC is improved to solve the given problem. The objective of this project is to improve the behaviour of the simulation with communication and reasoning process following a proposal.

B. Repast API

Repast Simphony, is a tightly integrated, richly interactive, cross platform Java-based modeling system that runs under Microsoft Windows, Apple Mac OS X, and Linux. It supports the development of extremely flexible models of interacting agents for use on workstations and computing clusters. Repast Simphony models can be developed in several different forms including the ReLogo dialect of Logo, point-and-click statecharts, Groovy, or Java, all of which can be fluidly interleaved. I've used and implemented the Repast API written in Java for the whole of the project. Repast has multiple implementations in several languages and built-in adaptive features, such as genetic algorithms and regression, but the improved DARPC is implemented in Java.
There are various features of Repast, which are namely,

— variety of agents and examples.
— fully object oriented.
— fully concurent discrete event scheduler.
— built-in simulation results logging and graphing tools.
— allows users to dynamically access and modify agents and model at run time.
— libraries for genetic algorithms, neural networks, etc.
— built-in systems dynamics modeling.
— social network modeling tools.
— integrated geographical information systems (GIS) support
— implemented in Java, C, etc.
— supports Java, C, Managed C++, Visual Basic.Net, Managed Lisp, Managed Prolog, and Python scripting, etc.
— is available on virtually all modern computing platforms.

Multi-Agent Systems

A Multi-Agent System is a system constructed from various autonomous elements interacting and reacting with one another. They are called Agents. In simulation, the agents are used to simulate various different elements. These could be society, organism, machine, person or any other active element. n a multi-agent system, an agent is represented by a software program or algorithm. This program contains in itself all rules of agents behavior. The purpose of models could be simulation of social phenomena like transportation, market failures, cooperation and escalation and spreading of conflicts. In Agent based social systems, agents Emergence in context of social simulation In agent based simulations we can observe phenomenon, when model based on simple rules results in very complex dynamics. This phenomenon is related to emergence and one of recent topic of social science is concept of emerging behavior in social science. They're highly important and useful in understanding the basic aspects of social phenomenon, prediction and research, testing and formulation of the hypothesis.
"Agent" is observed as an Java object with a special feature of being able to interact with it's surroundings and being able to iterate itself over with different values on a response, by 'responding' to the surrounding by changing it's value or behaviour. This is traditionally very different with the concept of an "Object" in Multi-Agent Systems, which is also an Java object but misses the features of being interactive with the surroundings or being 'intelligent'.

C. Literature Survey

Vehicle Routing Problems have been studied by researchers for well over 40 years, resulting into the development of various algorithms to solve the problem. But, the problem is an NP-Hard problem (Toth and Vigo, 2002) considering the fact that as the number of problem size as well as the number of variable increases in a VRP, the complexity of computation will also increase.

Common Approaches

There are majorly two common approaches involved to solve a DARP. The first approach will be to find an optimal fleet size and capacity configuration to satisfy all the customer requests. The second approach will revolve around the idea of finding the value of how many customer requests can be fulfilled by having a fixed fleet size (Cordeau and Laporte, 2007). DARP problems can be considered in the sub-domain of the Pickup and Delivery problems, where one of the major objective is to minimize the cost function (Berbeglia, 2007). The difference between DARP problems and Pick-up and delivery problems is the time constrained intervals included in which the service of each customer should be taken place by the vehicle. So, in DARP, a user inconvenience (delay in the time of getting a vehicle) is considered to compute the quality of the service. This makes DARP an harder problem to solve. In the event, if strict time windows are used, the resulting cost of transportation will be very high. Similarly, having well optimized routes can result into the customers having a lot of waiting time. The cost elements used in the objective function commonly relate to the fixed vehicle costs for each kilometre travelled, and the cost per minute for servicing customers linked to the drivers wages (Hansen, 2010). Service quality can be determined by making a comparison between the direct and excess ride times a customer spends travelling to his destination, and the time spent waiting before departure (Cordeau and Laporte, 2007). The DARP can be further classified into either a single capacity or multi-capacity vehicle problem. The fleet of vehicles can either have homogeneous or heterogeneous features. Features can relate to the vehicles' capacities or possible seating configurations. While most applications of DARP consider a homogenous vehicle fleet, Madsen et al. (1995) considered a multidimensional vehicle capacity requirement. Vehicles had a combination of ordinary seats, lying seats, children seats, wheel chair places and a number of bed places. The use of a heterogeneous fleet of vehicles increased the complexity of the DARP, as the seating requirements of customers form an additional strict customer service constraint. Industry applications of the DARP have been used in Copenhagen and Belgium (see Madsen etal., 1995, and Rekiek et al., 2006.) to transport the handicapped. In the USA there are approximately 1500 demand response transit systems to service the needs of customers in rural areas, with at least 400 such services existing in urban districts (Transit Cooperative Research Programme, 2009).
Thus to solve a DARP problem, following key decisions are required,

1. Determining how to cluster the customers to be serviced by the same vehicle.
2. Sequencing the servicing of customers into a vehicle route within their assigned cluster.
3. Scheduling the customer pick-up and drop-off events for each vehicle (Cordeau and Laporte, 2003).

To make these decisions in the case of DARP problem, there are two dominant groups of research methods being used, which are,

1. Insertion heuristics, whereby customers are optimally inserted into an initial route, making the solution quality reliant on the sequence in which customers are visited.
2. Cluster-first and route-second heuristics. This approach partitions all the customers into a number of subsets, equivalent to the number of vehicles available to service the customers. Optimal routes are then determined for each specific vehicle to service its customer subset.

The quality of the solution obtained relies on the partitioning of customers into vehicle subsets (Baugh et al., 1998).

Vehicle Scheduling

The static DARP can have difficulty adhering to pre-established schedules due to time varying factors, such as irregular customer demand or traffic congestion which can affect the execution of a set schedule. In general, given a sequence of

$$i_1, i_2, \ldots, i_q$$

nodes to be visited, the vehicle scheduling problem can be formulated as :

$$\sum_{i=1}^{q} g_i B_i$$

such that,

$$B_i + d_i + t_i, i + 1 \leq B_i + 1$$

where ,

$$\forall i = 1, 2, 3, ..., q - 1$$

and

$$e_i \leq B_i \leq l_i$$

where,

$$\forall i = 1, 2, 3, ..., q$$

Where giBi is a convex function that defines a time window

$$[e_i, l_i]$$

for each customer node i N. There exists a non-negative load qi and service duration di. Travel time tij is associated with each arc (i, j) A between customer nodes (Cordeau et al., 2007). Within a DARP of m vehicles, there exists an equivalent amount of m single vehicle routing subproblems. Each individual vehicle is concerned about how best to service its own customers in its designated customer cluster. Sexton and Bodin (1985) propose a two-step scheduling method that uses Benders Decomposition to solve a DARP. The first step involves partitioning customers into vehicle clusters. The cluster 10 are then each solved as a single vehicle routing problem (SVRP), which aims to minimise the total customer schedule inconvenience defined by the function :

$$Totalinconvenienceoftheschedule = \sum totalcustomerinconvenience$$

where

$$Totalcustomerinconvenience = \sum (excessridetime + deliverytimedeviation)$$

The excess ride time is the difference between the actual ride time and the direct ride time. The delivery time deviation is the difference between the desired delivery time by the customer and the actual delivery time. Using a weighted objective function that minimises customer inconvenience, Sexton and Bodin (1985) found the dual of the scheduling problem can be interpreted as a Network Flow Problem. This type of problem could then be solved very efficiently as a linear programming problem to determine an optimal vehicle schedule. Once the scheduling sequence has been determined, the actual arrival and departure times of vehicles at customer locations must be established. This is to ensure time window constraints are adhered to and route durations are minimised. Cordeau and Laporte (2003) present a scheduling programming heuristic that determines the required departure time at which vehicle k must leave the depot, to minimise the total route duration and unnecessary waiting time.

Table 1 – Comparison of algorithm approaches

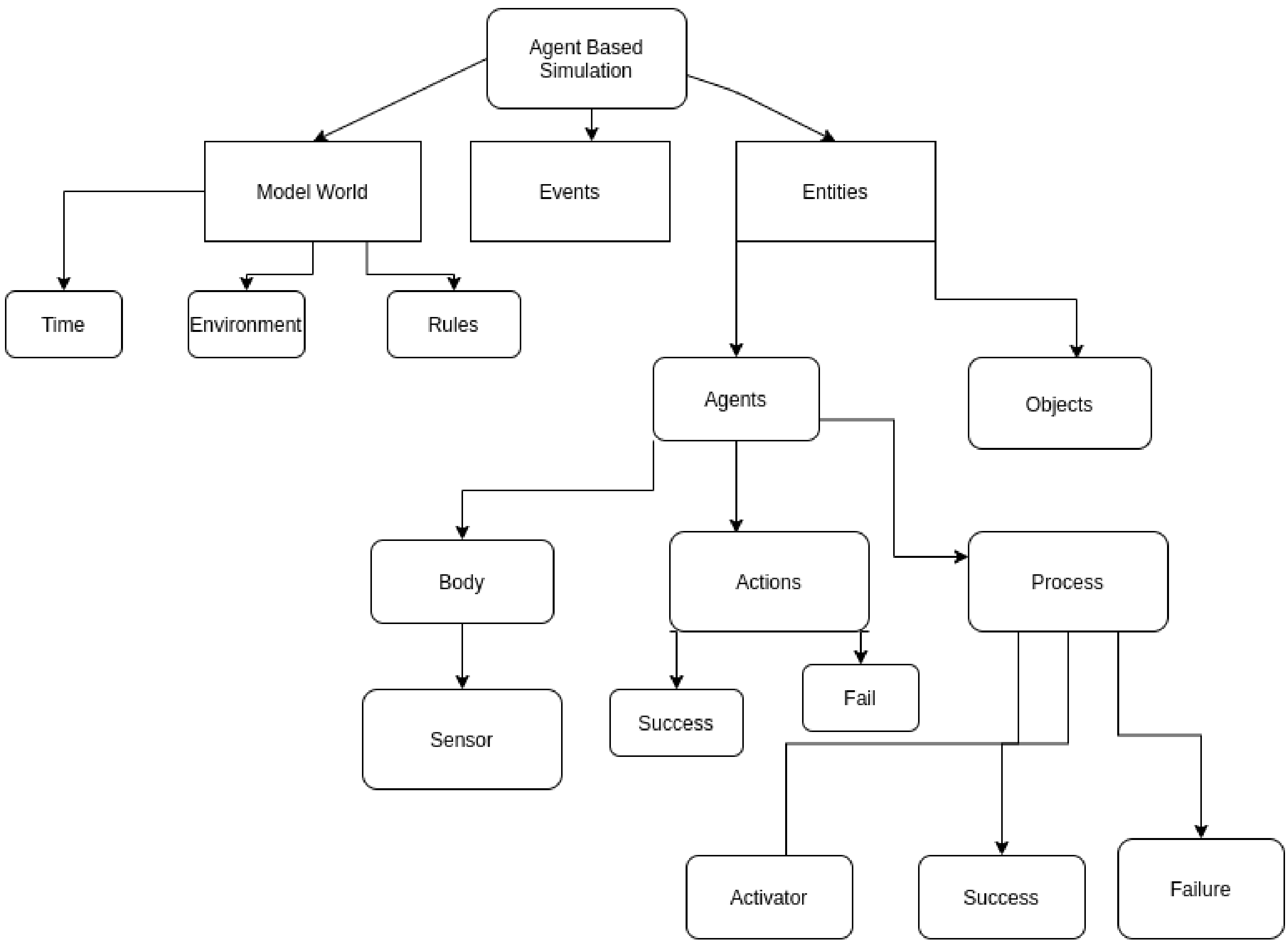
Author	**Jorgensen et al. (2007)**	**Cubillos et al. (2009)**
Objective Function	Transport and inadequate service cost	Vehicle travel, waiting, excess travel time.
Time Windows	Soft constraint	Hard Constraint
Routing	Genetic Algorithm	Genetic Algorithm
Clustering	Nearest Neighbour	Greedy Insertion

Agent Based Simulation

Agent Based Simulation has gained a lot of traction and importance in the recent years in accordance to the importance in simulating very complex systems. There is limited literature in regards to the application of the Agent Based Simulation. According to Segfried et al. (2009), an ABS is majorly characterized by,

1. Agents operating in an environment that is beyond their control
2. A simulated environment forms part of the model by providing the required infrastructure. The simulated environment may perform auxiliary tasks or assist with the interaction between agents
3. A step-wise execution approach where all agent actions' occur sequentially, with no event occurring in between the discrete event steps

The major elements which constitute the ABS are shown in the figure below,



During the running of time in simulations, various time subsets, specific to agents' may exist, provided that simulation time remains totally ordered. The model world environment denotes the common space that agents will function in. Events in an ABS are the set of all possible actions that may occur (e.g. customer pick-up or dropoff). Depending on the definition of the ABS, events may be either dependent or independent of time and may occur concurrently to one another, or may trigger another action. An agent in an agent based simulation is very different from that of an object, and that is because,

1. It has a physical dimension which forms its body, and is able to sense and interpret its environment
2. It is capable of performing multiple actions that consume a specific amount of time to be performed. Actions can be triggered by a sensor, with the outcome of an action deemed to be either a success or failure
3. An agent is closely interlinked with action processes. One action may activate other agent actions that form a process, with the possibility of actions occurring concurrently. A process would continue to run until it reaches a successful outcome or it is terminated due to a failure (Segfried et al. 2009).

Limitations of Agent Based Simulation

ABS is not well known outside the narrow ABS community, with limited application of the method beyond the purposes of scientific, educational and experimental use. ABS is still considered to be in the early stages of development, with no widely accepted and proven methodology existing for the development of an ABS. The complexity required to construct an ABS is considered vast, given that it is not based on traditional mathematical computations which can be performed relatively quickly using computers. In particular the computational requirements of an ABS can be extremely intensive as the number of agents increase. This is due to the design of an ABS which usually imposes communication requirement between agents which is not scaled linearly, drastically increasing the computations that are required to solve a problem (Salamon, 2011).

Conclusion from the review

There has been substantial literature present to solve the Vehicle Routing Problem, but the application is limited, limiting itself to scientific, educational and experimental use. Agent Based Simulation can be still considered to be in the early stages of development. The use of meta heuristics methods has also dominated the solving of the DARP, with no application found in literature of agent based simulation to solve the DARP. The field of agent based simulation is comparatively new, with a limited amount of literature being found applied to Vehicle Routing Problems.

D. Approach to solve DARP

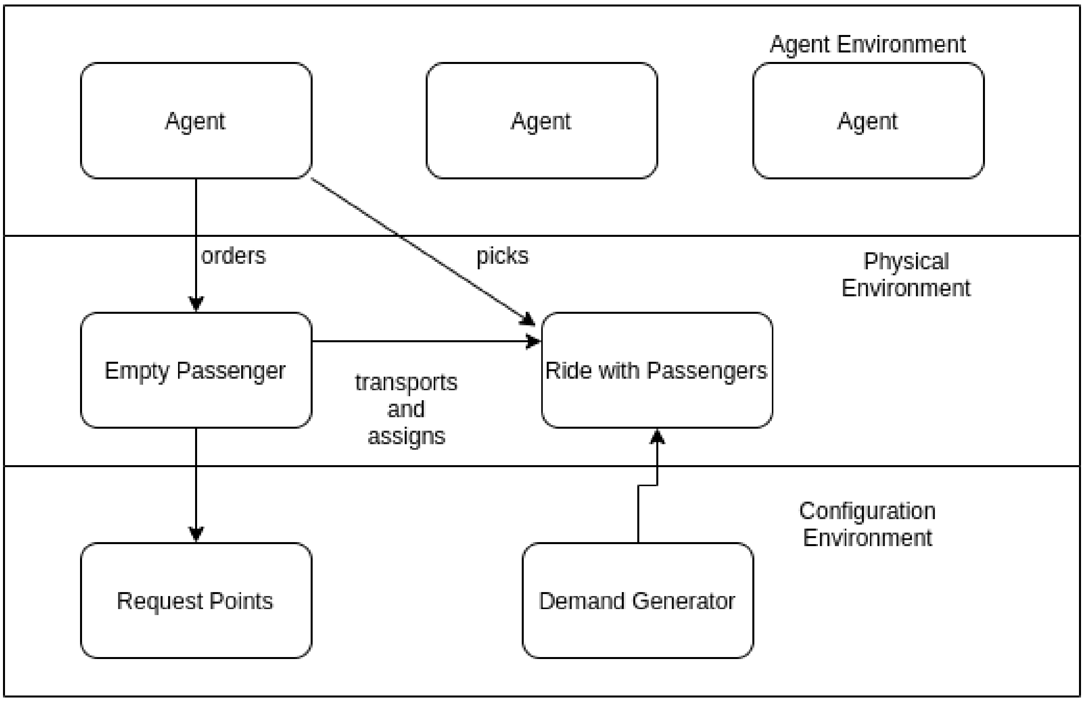
The main objective of the report is to improve the existing code that is present and has been given, and to improve it to increase some of it's features. This means that, we have a 'basicDialARide' package and to improve it. These improvements can be inform of increase of 'capabilities' of the simulation model.
The basicDialARide package has various features, namely,

1. Sending message back to the agent, returning information with fields as subject, performative, the content and the sender.
2. Scheduled method to run automatically the askARide function.
3. Agent gets location and returns the client information such as id, ride requested and the request.

4. Request class to give out requests to get a ride.
5. A context builder class with defined grid size and parameters, also contains the random string builder.
6. A scheduled method to make the vehicle move, pickup and drop off. This method also provides a method to calculate the travel time estimation.
7. A method which will change the color of the agent when it interacts with a client.

Improvements to the code.

The code was improved by adding two more classes regarding the Car and the Traffic Light in the simulation. These classes are supposed to provide the feature of improving the simulation by providing traffic lights to keep the agents in check, as well as random cars which aren't agents (normal 'not artificially intelligent' cars).



Schematic Display of the Simulation Model.

Implementing a fitness function : Fitness functions are, a type of objective functions which are used to summarize as a single feature of merit, of how close a given design solution is to achieving the set aims. They're widely used in optimizing simulations. In many cases, a multi-objective goal exists, requiring a multi objective definition of the problem. The main goal of the problem was to implement a mechanism to serve as many clients, and by also respecting the TotalCustomerInconvenience time that determines the quality of the service as well as the simulation model.

We define a fitness function which is a multiplication of the ratio of the rides met to the total clients and the averaged amount of the instances when the client didn't meet the ride. The fitness function is, however lacking the capability of being interpreted.

Solution to the questions of section 2 of the DARP Subject.pdf file.

Q1. Agents : What are agents and their behaviors ?
-> Agent is the ride, as it interacts with itself and the surrounding rides in the same locality to assign the closest ride to the client, and providing a return function with the message containing

the information about the ride as well as the client.

Q2. Environment : What are it's properties ?
-> Environment is the portion surrounding the agent system simulation model. As described in the schematic display of the simulation model, Physical environment contains the riders who are supposed to 'dial' a ride. The other riders as well as the other cars in the environment are the functional component of the environment, as they may interact with the simulation model on their wish. There are various other non-functional components with properties that do not interact with the simulation model, and thus are of no use. These are for example, the road (which I have tried to implement), the pedestrian and other natural conditions. There are also non-functional cars which are normal cars and are also part of the environment.

Q3. Interaction : How agents influence behaviors of other ?
-> Interaction between agent takes place, as various functions included in the builder function, agents don't take a ride if another agent is closer to the customer as they are. Thus, the constant interaction provides a cost minimizing feature to the whole simulation. There is no indirect interaction, as all the agents know about the other agent in the proximity.

Q4. Organization : What rules the behavior of agents ?
-> Agents have no defined permission in the simulation, they just have one obligation and that is to keep the customer inconvenience time as minimum as possible to provide the ride.

Difficulties Encountered.

1. Managing the grid size was a difficult task, as to know the optimum amount of agents and the riders.
2. Executing the simulation, there were difficulties I have encountered.
3. Time Management, apologies for the late submission.
4. Formatting the report in LaTeXwas a bit difficult and I didn't do well, I'm new to LaTeXand I'm still learning it. Apologies for any inconvenience.

E. REFERENCES

1. Collier, N., North, M. (2013). Parallel agent-based simulation with Repast for High Performance Computing. SIMULATION, 89(10), 1215–1235.
2. North, MJ, NT Collier, J Ozik, E Tatara, M Altaweel, CM Macal, M Bragen, and P Sydelko, "Complex Adaptive Systems Modeling with Repast Simphony", Complex Adaptive Systems Modeling, Springer, Heidelberg, FRG (2013).
3. Toth, Paolo Vigo, Daniele. (2002). The Vehicle Routing Problem. 10.1137/1.9780898718515.
4. Cordeau, JF., Laporte, G. The dial-a-ride problem : models and algorithms. Ann Oper Res 153, 29–46 (2007). https ://doi.org/10.1007/s10479-007-0170-8
5. Kuschel, Torben. (2008). Two-Commodity Network Flow Formulations for Vehicle Routing Problems with Simultaneous Pickup Delivery and a Many-to-Many Structure. Informations- und Kommunikationssysteme in Supply Chain Management, Logistik und Transport,. 5. 153-169.
6. Khanna S., Cleaver T., Sattar A., Hansen D., Stantic B. (2012) Multiagent Based Scheduling of Elective Surgery. In : Desai N., Liu A., Winikoff M. (eds) Principles and Practice of Multi-Agent Systems. PRIMA 2010. Lecture Notes in Computer Science, vol 7057. Springer, Berlin, Heidelberg. https ://doi.org/10.1007/978-3-642-25920-36

7. Sophie N. Parragh, Karl F. Doerner, and Richard F. Hartl. "A survey on pickup and delivery problems" Journal für Betriebswirtschaft, vol. 58, no. 2, 2008. doi :10.1007/s11301-008-0036-4
8. C. M. Baugh, A. J. Benson, S. Cole, C. S. Frenk, C. G. Lacey, Modelling the evolution of galaxy clustering, Monthly Notices of the Royal Astronomical Society, Volume 305, Issue 1, May 1999, Pages L21–L25, https ://doi.org/10.1046/j.1365-8711.1999.02590.x
9. Sexton, Thomas R. and Bodin, Lawrence D., (1985), Optimizing Single Vehicle Many-to-Many Operations with Desired Delivery Times : II. Routing, Transportation Science, 19, issue 4, p. 411-435, https ://EconPapers.repec.org/RePEc :inm :ortrsc :v :19 :y :1985 :i :4 :p :411-435.
10. Mikkelsen, Lars Jørgensen, Bo. (2012). APPLICATION OF MULTI-AGENT SYSTEMS IN OFFSHORE OIL AND GAS PRODUCTION.
11. Cubillos, Claudio Gaete, Sandra Guidi-Polanco, Franco Demartini, Claudio. (2007). Design of a Multiagent Solution for Demand-Responsive Transportation. 797-804. 10.1007/978-3-540-74782-671.
12. Tomas Salamon. 2011. Design of Agent-Based Models. Eva Tomas Bruckner Publishing, Repin, CZE.