ӘДЕБИЕТ

[1] Кремер А.Н. Транспортная задача.– Новосибирск, 2011. – 233с.

[2] Максимов Н.П. Имитационное моделирование. - М.: Наука, 2005. - 145 с.

[3] Белов А.М. Принятие решений в сложных ситуациях. – М.: Высшая школа, 2013. – 301с.

[4] Колмагоров А.Н. Введение в теорию обоснованного выбора вариантов. – М.: Наука, 2015. – 364с.

[5] Черноруцкий И.Г. В. Методы принятия решений. – Санкт - Петербург, 2015. - 408с.

[6] Петров А.А. Транспортная задача. - М.: Наука, 2018. – 262с.

Букенова И.Н., Букенов Г.С.

Исследование методики иммитационного моделирования для оценки зависимости решения транспортной задачи от погрешности исходных данных

Резюме. Городские транспортные сети с позиции решения проблемы управления относятся к классу систем с пространственно распределенными объектами, что ограничивает возможность использования для их исследования традиционных математических методов моделирования и оптимизации. Все это требует разработки современного математического аппарата, ориентированного на реализацию в рамках интегрированных информационных компьютерных систем с активным включением сервисных компонент, обеспечивающих эффективный процесс взаимодействия при принятии управленческих решений.

Многие задачи организации перевозок, экономические задачи, распределение ресурсов сводятся к транспортной задаче.

При этом в существующих работах мало уделено внимание оценки влияния погрешности исходных данных на оптимальное решение транспортной задачи, хотя при решении практических задач этот фактор имеет не маловажную роль. В связи с этим в этой работе рассматривается методика оценки влияния погрешности исходных данных на решение транспортной задачи с применением имитационного моделирования.

В данной статье представлен алгоритм метода имитационного моделирования. Имитация проведена для оценки точности решения транспортной задачи в зависимости от погрешности исходных данных. В результате получены средние значения, среднеквадратические отклонения. А также доверительные интервалы целевой функции и решении транспортной задачи, когда исходные данные являются неопределенными. Предлагаемое специальное программное обеспечение может быть использовано как для решения прикладных транспортных задач, так и при проведении научных исследований и в учебном процессе.

Ключевые слова. Имитационное моделирование, транспортная задача, погрешность исходных данных, доверительные интервалы, целевая функция, оценка точности решения.

УДК 004.89

L.M. Alimzhanova, V.I. Karyukin
(al-Farabi Kazakh National University, Almaty, Republic of Kazakhstan)
Email: lauralim01@gmail.com, vladislav.karyukin@kaznu.kz

A CLASSIFICATION MODEL BASED ON DECISION-MAKING PROCESSES

Abstract. The article is devoted to the formation of a classification of decision-making systems from the simplest having an algorithm of sequential operations to complex systems based on control of an object taking into account factors of the environment that are changed during a process of the controlled object’s work. Factors could be static and functional. It significantly affects the decision-making trajectory, which makes this system difficult to predict for the developers. These requirements of an intellectual system suppose absence of an explicit algorithm.

Keywords: an intellectual system, decision-making processes, a linear algorithm, a conditional algorithm, a rational system

Introduction

Currently, there is no clear differentiation of decision-making models from simple to complex ones. This study proposes a version of the classification of algorithms based on the complexity of decision-making processes. A structural description of the existing decision-making models is given below.

1. A linear structure of sequential simple operations;
2. A simple algorithm with branches and cycles;
3. An algorithm that is based on the use of a database of ready solutions that are unchanged;
4. An expert system that allows you to make elementary decisions from an unchanged database of ready solutions. External factors and rules also do not change;

5. Systems that accumulate data, i.e. the database of ready solutions increases in time, which subsequently leads to a change in the database according to the rules.

Methods

Decision-making models are represented by the categories described below.

1. Systems based on simple algorithms

1.1 Basic linear algorithms

The first computer programs realized the concept of structured programming.

All instructions were executed one by one. When a previous instruction was completed, the next one started to work. Structured programming showed itself very useful and trivial for solving tasks that do not have many conditions and circumstanced required to be analyzed. This simple algorithm includes only a sequence of steps that have to be completed to receive a specific result. Figure 1 illustrates processes in a system that are implemented in a chain of steps. Each process contains some operations that are required to be completed. After completing the first process, the second one starts its operations. This logic continues until the end of the program. This trivial program does not imply any branches and different variants.

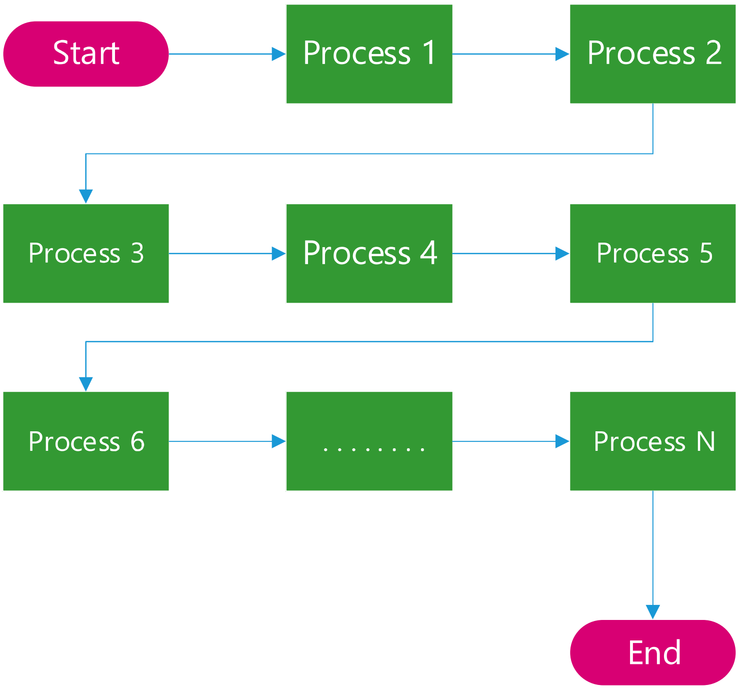


Figure 1. A linear algorithm

1.2 Conditional algorithms

As linear algorithms do not allow any strategic decisions, and they are practically useless in modern systems, they had to be modified to include functionality of conditions and transitions. It was realized by an addition of branches, cycles, conditional and unconditional transitions. Branch statements are formed by «if…then…else» or «case» operators that will provide different kinds of choices for users of programs. Iterative statements «for, while, repeat» provide functionality to receive an estimated result after multiple passes. The work of this kind of algorithms is in Figure 2. These blocks include a beginning and an end of the program, rectangular shapes show activity blocks and diamond shapes illustrate conditions that lead to different ways towards the final result.

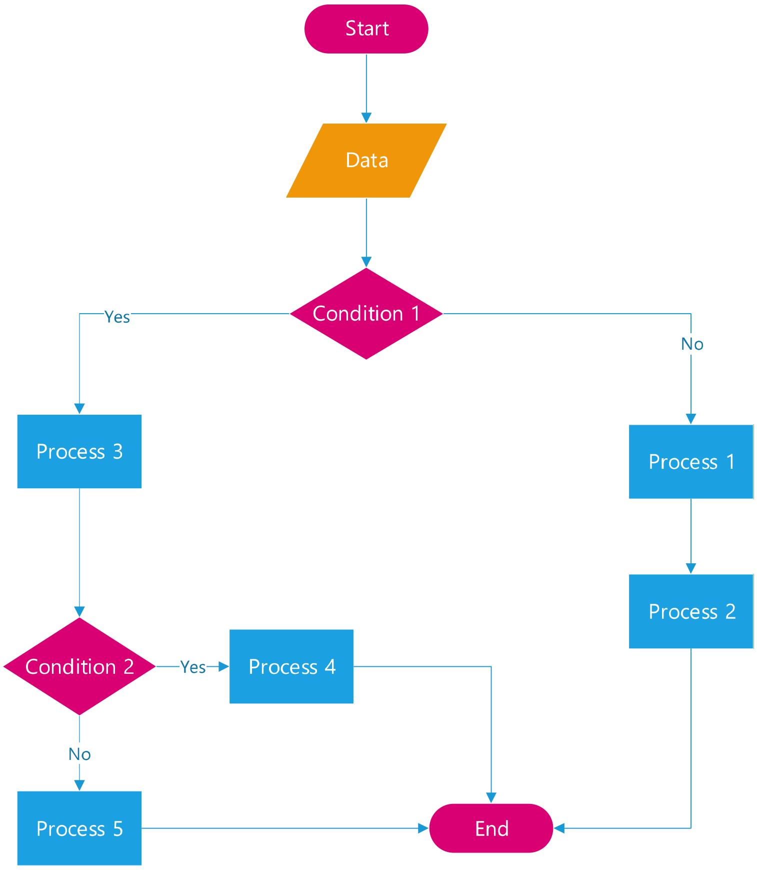


Figure 2. A conditional algorithm

2. Rational systems

2.1 Systems with the non-changing parameters of the environment

Although stated above simple algorithms allow to fulfill some specific tasks, they are very restricted in functionality. It is just a direct sequence of actions and variants without any analysis of situations. Modern systems have to provide the flexible functionality where the environment and future steps are analyzed in advance.

The concept of rationality in the Artificial Intelligence field comes to help here. An agent is everyone or anything that performs actions. The main difference between these kinds of programs and programs mentioned previously is execution of non-straightforward steps. At the same time, the agent always perceives the environment where it operates via various kinds of sensors and actuators [1].

2.2 Systems with the changing parameters of the environment

In the previous system we have the parameters of robots and environment always stable and non-changing. There are many cases when external factors influence the work of robots, and they need to adapt to them. Robots have to make decisions that will allow to show the same performance measure they had before the change of factors [2]. The trajectory plan is to be divided into small subtasks which execution leads to the final goal of the system.

Taking into account the result of the article [3], it is required to highlight that the model presented there is described inaccurately by the following parameters:

1. There are no basic hypotheses for the formation of the conceptual model;
2. Since there are no corrections at individual stages, no mechanism of the influence of external factors on the trajectory of performing a complex task in the diagram is presented for the conceptual task. A trajectory of each consequent stage is fully dependent on the results of the previous stage. Factors of external influence are absent, but the scheme intends that they have to exist;
3. An elementary mathematical description of the model is absent that raises a number of questions about the correctness of the described model. The proposed work gives a detailed description of the model and discloses principles of the model in a more correct form.

The hypotheses on the basis of which this conceptual model is proposed are described below:

1. In the first case, an agent makes the best decision at the beginning taking into account the amount of relevant information received;
2. In the second case, an agent is also influenced by external factors at the intermediate points, and its trajectory changes correspondingly;
3. In the third case, external factors change values at the intermediate points, and it influences the behavior of the agent even more that in the second case.

The rational behavior is expressed with the use of the following mathematical equations.

$$\begin{cases} p_0 = q_0(S_0, x_1, x_2, \dots, x_j, \dots, x_m), \\ p_1 = q_1(S_1), where\ S_1 \equiv p_0 \\ \dots \\ p_i = q_i(S_i), where\ S_i \equiv p_{i-1} \\ \dots \\ p_n = q_n(S_n), where\ S_n \equiv p_{n-1} \end{cases} \tag{1}$$

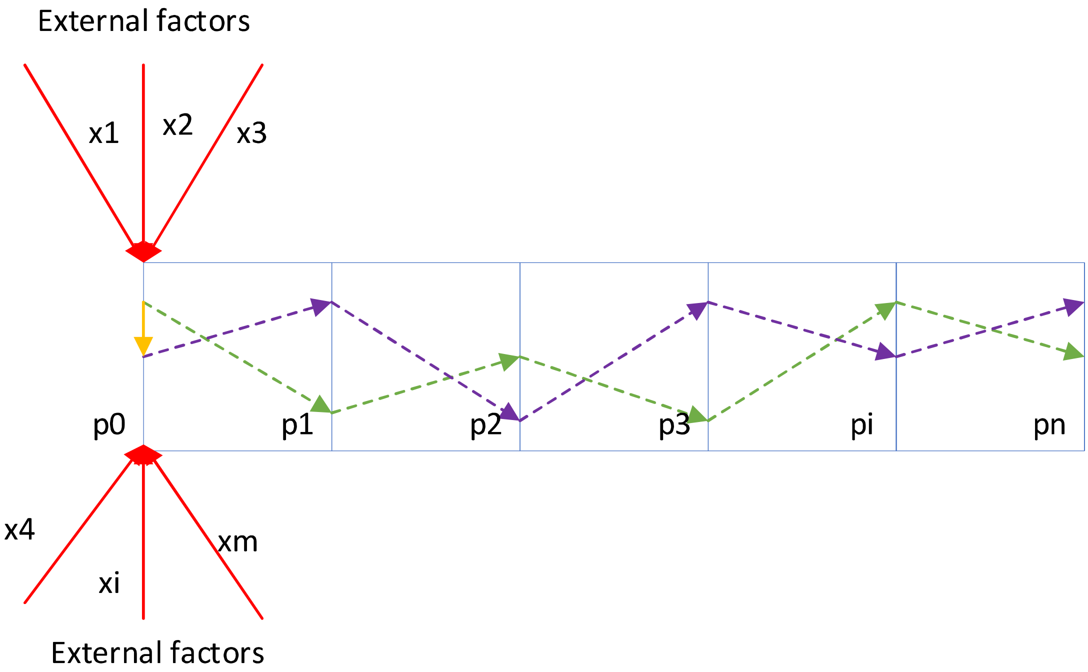


Figure 3. Model 1

In the first model, a list of initial positions is defined as $S_0, S_1, S_2, \dots, S_m$. The start position is influenced by external factors $(x_0, x_1, \dots, x_j, \dots, x_m)$. Every next step is the functional result of the previous step.

In the second model, a correction function that depends on the external factors at the control points exists. There will be a bias at the intermediate steps of the trajectory. So external factors can influence agents only at those points, and it is revealed inside a correction function.

$$\begin{cases} p_0 = q_0(S_0), S_0 = f\left(x_{01}, x_{02}, \dots, x_{0j}, \dots, x_{0m}\right) \\ p_1 = q_1(S_1), where\ S_1 \neq p_0, S_1 = \varphi_1(p_0, x_{11}, x_{12}, \dots, x_{1j}, \dots, x_{1m}) \\ \dots \\ p_i = q_i(S_i), where\ S_i \neq p_{i-1}, S_i = \varphi_i(p_{i-1}, x_{i1}, x_{i2}, \dots, x_{ij}, \dots, x_{im}) \\ \dots \\ p_n = q_n(S_n), where\ S_n \neq p_{n-1}, S_n = \varphi_n(p_{n-1}, x_{n1}, x_{n2}, \dots, x_{nj}, \dots, x_{nm}) \end{cases} \tag{2}$$

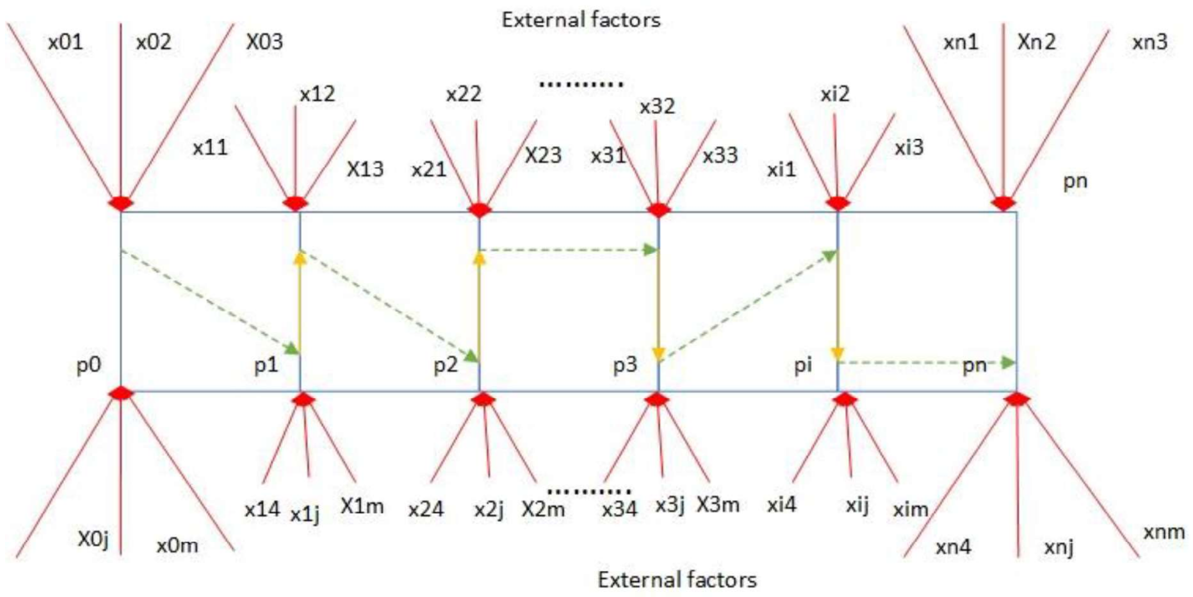


Figure 4. Model 2

In the third model, a correction function shows the change of external factors at the intermediate steps of the trajectory over time.

$$\begin{cases} p_1 = g_1(S_1, x_{01},\ x_{02}, \dots, x_{0j}, \dots, x_{0m}), S_1 = f\big(x_{01},\ x_{02}, \dots, x_{0j}, \dots, x_{0m}, t\big) \\ p_2 = \ g_2(S_2, x_{11},\ x_{12}, \dots, x_{1j}, \dots, x_{1m}), \\ S_2 \neq p_1, S_2 = \varphi_1(p_1, x_{11},\ x_{12}, \dots, x_{1j}, \dots, x_{1m}, t) \\ \dots \\ p_i = \ g_i(S_i, x_{i1},\ x_{i2}, \dots, x_{ij}, \dots, x_{im}), \\ S_i \neq p_{i-1}, S_i = \varphi_{i-1}(p_{i-1}, x_{i1},\ x_{i2}, \dots, x_{ij}, \dots, x_{im}, t) \\ \dots \\ p_n = \ g_n(S_n, x_{n1},\ x_{n2}, \dots, x_{nj}, \dots, x_{nm}), \\ S_n \neq p_{n-1}, S_n = \varphi_{n-1}(p_{n-1}, x_{n1},\ x_{n2}, \dots, x_{nj}, \dots, x_{nm}, t) \end{cases} \qquad (3)$$

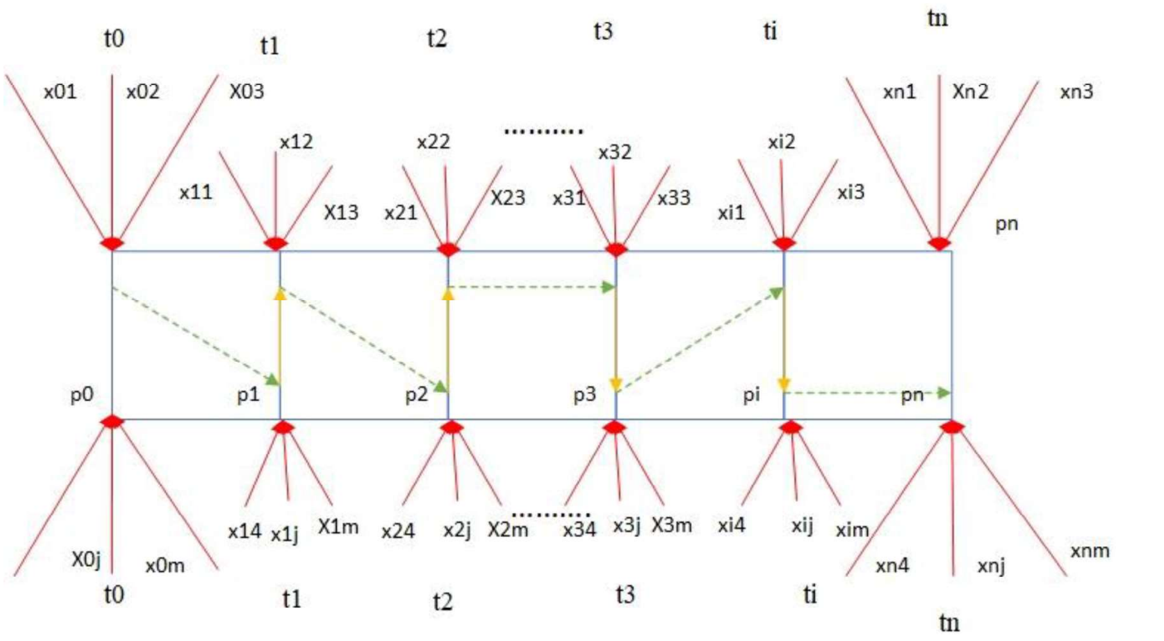
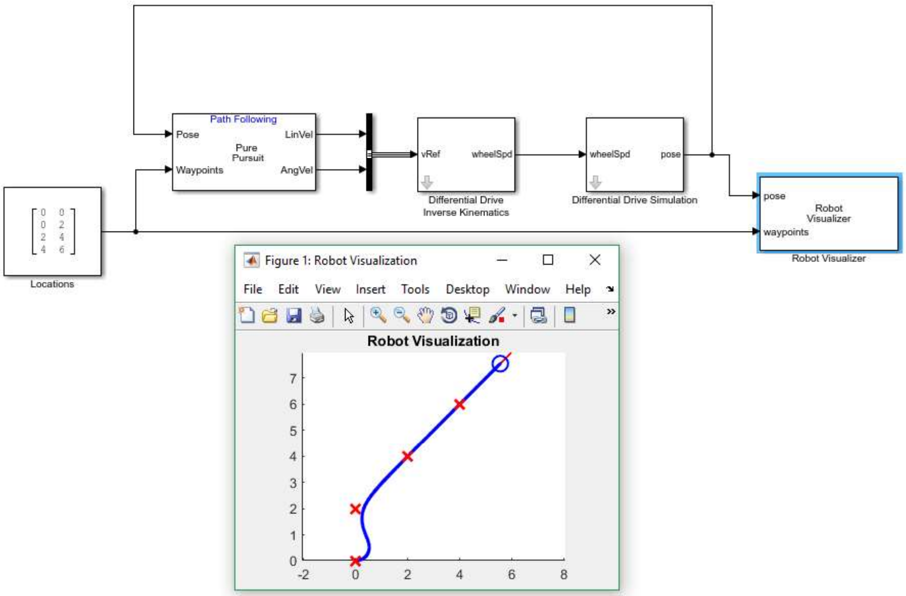


Figure 5. Model 3

Realization

In the robotic system toolbox in MATLAB, the linear algorithm may be represented by a vehicle moving from the start point to the end point. A trajectory of the movement is quite simple including only intermediate points along its path. At the beginning, it is necessary to set up linear and angular velocity values that define how fast an object will be moving towards the end point [4-7]. Points of the trajectory are described by an input matrix.

The following parameters have been used in the model: wheel radius – 0.1, wheel base – 0.5, linear velocity – 0.5, angular velocity – 1.0



A moving object 1

To demonstrate a variety of decisions made by robots, it is worth to use an example of the point detection algorithm in MATLAB. A point is set up on the map, and a wheel drive robot uses its scanner to find this point and move towards it. If the point is not found by the robot, it does not try to reach it. Otherwise, when the point is detected, the robot is going to move to that place [8-9]. This example is presented on the image below.

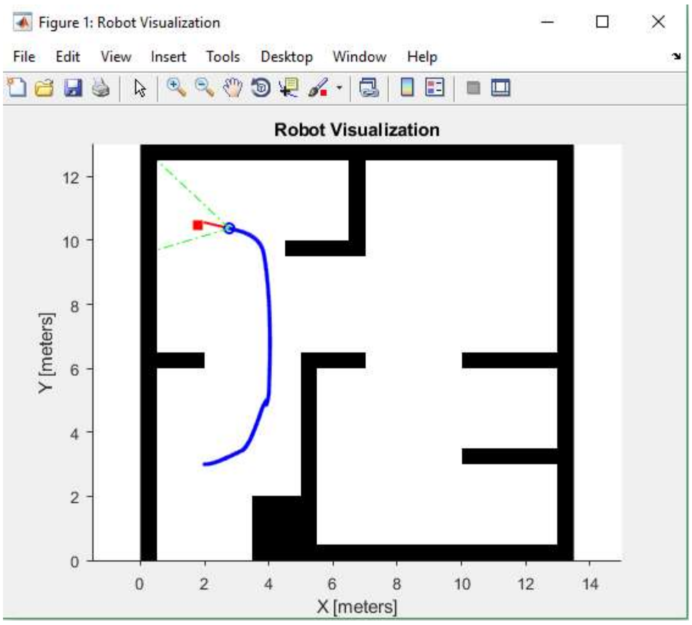


Figure 1: Robot Visualization
File Edit View Insert Tools Desktop Window Help
Robot Visualization
Y [meters]
12
10
8
6
4
2
0
0
2
4
6
8
10
12
14
X [meters]

A moving object 2

Knowledge gathering system

One efficient way of keeping data about actions taken by agents is the use lists that bind facts in the environment with new rules an agent has to take into account executing its actions. A widespread example of the agent’s rational work is the Wumpus game [1]. This game includes a hero (agent) who enters a cave where he needs to find gold and kill the terrifying monster Wumpus. In addition, a cave contains deep pits that are dangerous for a hero because he can fall there and die.

The cave is represented by a board of cells. The hero moves horizontally and vertically. If a cell contains a cave, there is wind in nearby cells. The hero can also smell Wumpus in the cells next to the monster’s location. A hero has only one arrow to kill the Wumpus with it. In order to murder the monster, the hero has to know his location for sure. As the hero has no prior knowledge about a structure of the cave, he needs to build it while he moves. Sensors of the agent provide him the following information:

- wind in the cells next to pits;
- stench in the cells near the Wumpus;
- bump when an agent collides with walls;
- glitter when there is gold that an agent has to find;
- scream when the Wumpus is killed by an arrow.

The agent has a list of rules initially, and additionally, he needs to fill its table of facts which it gets from the cave (Table 1).

Using accommodated facts and rules, the agent can successfully fulfill its task.

Table 1. **Agent’s work specifications**

Facts and rules	Steps of the agent
#facts and rules that we will use to make inferences *facts = []* *rules = [* *([('clean', '?x'), ('next', '?x', '?y')], [('no_wumpus', '?y')]),* *([('calm', '?x'), ('next', '?x', '?y')], [('solid', '?y')]),* *([('no_wumpus', '?x'), ('solid', '?x')], [('safe', '?x'),]),* *([('next', '?a', '?b'), ('next', '?b', '?c'), ('next', '?c', '?d'), ('next', '?d', '?a'), ('nasty', '?a'), ('no_wumpus', '?b'), ('nasty', '?c')], [('wumpus', '?d'),])]*	Taking a step Moving to Cell 21 There is a breeze running through this room Perception = (clean, breeze, bare, no_bump, quiet, Cell 21, Left, living, 1100) ********************************* Turing to face Left There is a breeze running through this room Perception = (clean, breeze, bare, no_bump, quiet, Cell 21, Left, living, 1100) ********************************* Taking a step Moving to Cell 11 Perception = (clean, calm, bare, no_bump, quiet, Cell 11, Left, living, 1100) ********************************* Perception = (clean, calm, bare, no_bump, quiet, Cell 11, Left, won, 1100)

Conclusion

The classification model demonstrated a shift from basic sequential algorithms to complex decision-making systems where intelligent agents and robots are influenced by a large group of different factors. In this case, it is actual to thoroughly describe the change of behavior on different stages of work.

The intellectual part of the decision-making system is confirmed by the presence of an implicit algorithm for performing tasks by robotic systems based on the presented models. These models demonstrate a gradual complication of the task execution algorithm depending on the degree of influence of external factors. If in the first mathematically described model, factors influenced and were taken into account only at the initial stage, then in more complex models, external factors first affect the degree of correction of the execution of the task steps only at control points. In the final model, factors already begin to change the algorithms for the execution of the stages (subtasks) of the complex task themselves. Subsequently, the authors will develop this topic, taking more and more complex forms of changing the basic algorithm in order to bring the solution closer to a more intellectually advanced solution variant.

REFERENCES

[1] Stuart Russel, Peter Nordvig. Artificial Intelligence, a modern approach. – 3rd edition. – Prentice Hall. – Pages 1152

[2] Timotheus Kampik, Juan Carlos Nieves, Helena Lindgren. Explaining Sympathetic Actions of Rational Agents // Lecture Notes in Artificial Intelligence. – 2019. – Volume 11763. – Pages 59-76.

[3] Uskenbayeva R., Kuandykov A., Kozhamzharova D., Duisebekova K., Alimzhanova L., Alimanova M., Sarbasova A., Adilzhanova, S. The development of conceptual model of creation of an intelligent system: Application on robotic systems // 1st IEEE International Conference on Robotic Computing, IRC 2017, Taichung, Taiwan. – 2017. – Pages 362-364.

[4] Lujak. M. A distributed coordination model for multi-robot box pushing // IFAC Proceedings. – 2010. – Volume 10. – Pages 120-125.

[5] Murray, R.M. Recent research in cooperative control of multivehicle systems // Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME. – 2007. – Volume 129 (5). – Pages 571-583.

[6] Shiyu Zhang, Andrea Maria Zanchettin, Renzo Villa, Shuling Dai. Real-time trajectory planning based on joint-decoupled optimization in human-robot interaction // Mechanism and Machine Theory. – 2020. – Volume 144.

[7] M.F.Laguna, J.R.Iglesias, Sebastián Gonçalves. Irrational behavior in the adoption of innovations // Physica A: Statistical Mechanics and its Applications. – 2019. – Volume 535.

[8] Haluk Ozakyo, Cenk Karaman, Zafer Bingul. Advanced robotics analysis toolbox for kinematic and dynamic design and analysis of high-DOF redundant serial manipulators // Computer Applications in Engineering Education. – 2019. – Volume 27, Issue 6. – Pages 1429-1452.

[9] Marian Lupascu, Sofia Hustiu, Adrian Burlacu, Marius Kloetzer. Path Planning for Autonomous Drones using 3D Rectangular Cuboid Decomposition // 23rd International Conference on System Theory, Control and Computing, ICSTCC 2019. – 2019. – Pages 119-124.

Алимжанова Л.М., Карюкин В.И.

Классификационная модель на основе принятия решений

Резюме. Статья описывает формирование классификационной модели системы принятия решений, начиная от простых алгоритмов последовательных операций до сложных систем, в основе которых лежит влияние факторов окружающей среды. Факторы могут как статическими, так и функциональными. Это существенно влияет на траекторию принятия решений, и система становится трудно предсказуемой для разработчиков. Данная особенность соответствует требованиям интеллектуальной системы и подчеркивает отсутствие явно заданного алгоритма.

Ключевые слова: интеллектуальная система, процессы принятия решений, линейный алгоритм, условный алгоритм, рациональная система.

Алимжанова Л.М., Карюкин В.И.

Шешімдерге негізделген жіктеу үлгісі

Түйіндеме Мақалада қоршаған ортаның факторларының әсеріне негізделген жүйелі операциялардың қарапайым алгоритмдерінен бастап күрделі жүйелеріне дейінгі шешімдер қабылдау жүйесінің классификациялық моделін қалыптастыру сипатталған. Факторлар статикалық та, функционалды да болуы мүмкін. Бұл шешім жолына айтарлықтай әсер етеді, ал әзірлеушілер үшін жүйені болжау қиынға соғады. Бұл функция зияткерлік жүйенің талаптарына сәйкес келеді және айқын алгоритмнің жоқтығын баса көрсетеді.

Түйінді сөздер: интеллектуалды жүйе, шешім қабылдау процестері, сызықтық алгоритм, шартты алгоритм, рационалды жүйе.

ӘОЖ 004.9

[1,2]F.U. Malikova, [1]G.A. Omarbekova

([1]Almaty Technological University, Almaty, Kazakhstan,
[2]Kazakh National University named after Al-Farabi, Almaty, Kazakhstan
E-mail: guldana.omarbekova@mail.ru)

IMPROVE AND OPTIMIZE WEBSITE INTEGRATED BOOTSTRAP 4 TEMPLATE IN DJANGO

Abstract. Combining the Bootstrap 4 template in Django, the web site development and optimization site (www.atu.kz site) provides an overview of the new ways to create a WEB site, methodologies of clarity and convenient delivery of information, peculiarities of Python, Django.

Website is a presentation of the history of our university, its services, its competitive advantages, products, terms of cooperation, etc. let alone talk about it. Developing Web site development technologies, developing web site