INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 12(6), 803-812, December 2017.

Intelligent Decision Support Algorithm Based on Self-Adaption Reasoning

G. Chen, Y. Jin, H. Wang, S. Cao

Guomin Chen, Yingwei Jin,
Huili Wang*, Shuo Cao
Dalian University of Technology
dr.chen@vip.126.com, jinyw67@dlut.edu.cn
*Corresponding author: huiliw@dlut.edu.cn
caoshuo@dlut.edu.cn

Abstract: This paper analyzes the logic deduction and reasoning techniques used in several intelligent decision support algorithms, and proposes a flexible planning method *GARIv* using fuzzy descriptive logic in media enterprise management. Combined with experiments, the above methods are illustrated in terms of effectiveness and feasibility. In the end, the challenges and possible solutions of intelligent decision support algorithms with self-adaption reasoning are discussed.
Keywords: intelligent decision support, propositional logic, non-monotonic logic, descriptive logic, fuzzy logic, automatic reasoning.

1 Introduction

Intelligent decision support (artificial intelligence planning), also known as automated planning (automated planning), is an important area of artificial intelligence research, and covers a knowledge representation, automatic reasoning, non-monotonic logic, human-computer interaction and cognition science and other areas of cross-disciplinary [6]. The first intelligent decision support research, which originated from the automatic reasoning and knowledge representation, in the 20th century, 90 years ago, has been the use of logical deduction method to be solved, with the focus on classical logic of reasoning. Mahmood [9]solves the problem of exponential space explosion in the process of knowledge representation, which draws the field of intelligent decision support more and more attention from researchers.

Intelligent decision support technology has been widely used in aerospace, robot control, logistics scheduling, game character design and system modeling. The results are obvious [8]. The US military forces in the Gulf War dynamic analysis and re-planning Tools DART is used in automated logistics planning and transportation scheduling, so that the scheduling work, which used to take a few weeks, now can be completed within a few hours. Intelligent decision support technology in the field of production scheduling and Mars Rover, Hubble Space Telescope and other aerospace fields. The application also shows its huge application prospects [11]. Especially in recent years, this huge application has also been demonstrated in the field of on-line education, such as Dalian Dragon Stone Internet EDU Service co, who employs the knowledge-based system for learning performance analysis and evaluation [7].

The International Planning Competition (IPC) has made great progress in the field of intelligent decision support technology, for which 5 sessions have been held continuously since 1998 [12]. The planning system of the previous three competitions mainly deals with the system performance, with the handling time of planning problems being the only criterion to measure the pros and cons of the system. In IPC-4 the consideration of complexity of the problem processing capacity is added to the probabilistic programming problem. In IPC-5 the Conformant planning problem is added. In the IPC development process, the scale of problem which needs to dealt with is gradually increased, so that the difficulty is gradually increased. The standard problem can describe the complicated and objective world more closely, enabling the intelligent decision support research to deal with complex practical problems.

The existing planning methods are divided into two main categories: fusion planning and fusion state space heuristic search planning [14]. By convergence transformation planning method, planning problems are converted into a number of classic problems (such as Propositional satisfiability problem, model checking problem, constraint satisfying problem, linear programming problem, theorem proving problem of non-classical logic, *etc.*), and solving the original programming problem indirectly through solving the transformed objective problem efficiently. In various fusion transformations planning, the logical deduction and reasoning techniques are used, more or less. The planning of the first class predicate logic is the source of these phenomena. It is very important to improve the efficiency of the planning method by using logical deduction and reasoning techniques.

This paper mainly studies the planning method and planning system of fusion logical deduction and reasoning technology. Firstly, the relevant concepts and problem description of intelligent decision support are introduced. Then the planning methods, which combine fusion logic deduction and reasoning technology are described, with a focus on the famous international planning system: flexible programming method of fusion fuzzy description logic. Finally, based on fuzzy descriptive logic, a flexible planning method is proposed and experimentally proved effective and feasible in the operation and management of media enterprises.

1.1 Intelligent decision support and fusion self-adaptive reasoning programming method

At present, intelligent decision support has not yet been defined uniformly, but it is generally believed that intelligent decision support is the process of finding the sequence of actions from the initial state to the target. According to Aleksander [1], intelligent decision support is defined as the "understanding and analysis of the surrounding environment. The goal of the realization of a number of alternative actions and the resources provided by the reasoning to develop a comprehensive action to achieve the goal sequence. Most of the media companies have witnessed an increasing online application, despite the combination of online and paper, which therefore leads to the wider use of computer-based adaptive technology.

1.2 Intelligent decision support problem

The description of the planning problem, usually based on the common planning domain description language (PDDL) [5], is similar to the representation of the first-order predicate logic, which adds conditional effects to the requirements of IPC, Numerical effects, probability effects, etc., leading to more detailed descriptions of the planning problem to be handled. Now the most common is PDDL3.2, which contains a description of the predicate and soft constraints and the description of the various logical relations.

And the abstract description of the planning problem is restricted to classical programming problems and non-classical programming problems. Non-classical programming problems mainly refer to conformant programming problems and flexible programming problems as related to this paper.

The problem of programming includes the domain definition and problem definition, which define the operation of the planning problem and the initial / target state respectively. A classical programming problem P is a triad (I, O, G), where I is the initial state, O is the operation set, and G is the target state set. The preconditions and the effects of each operation $o \in O$ are deterministic. Fusing the closed world hypothesis, all the propositions that I describe a state are explicitly depicted, and the true value of the proposition that is not present in the statement

I is false. That is the condition to be satisfied to describe the target state G. Example 1 is a partial description of a classical programming problem that fuses PDDL.

Example 1. (Logistics Planning Problem): The Logistics Planning Problem describes a logistical problem in which multiple parcels are transported to a designated location by loading, shipping, and unloading for multiple trucks and packages in several cities for a logistics package that contains two packages A, B , two locations L, P , and a truck R planning problem, the domain description and problem description is as follows:

Domain description:

$$\begin{array}{l} load(X,Y,Z) : PRE : at(X,Z), at(Y,Z); ADD : in(X,Y) : DEL : at(X,Z) \\ unload(X,Y,Z) : PRE : in(X,Y), at(Y,Z); ADD : at(X,Z); DEL : in(X,Y) \\ move(X,Y,Z) : PRE : at(X,Y), fuel(X); ADD : at(X,Z); DEL : at(X,Y), fuel(X) \end{array} \tag{1}$$

Problem description:

$$\begin{array}{l} Init : at(A,L), at(B,L), at(R,L), fuel(R) \\ Goal : at(A,P), at(B,P) \end{array} \tag{2}$$

Conformant programming problem is a triad , in which is the initial state set, is the operation set, and is the target state set. Here it is no longer a state, but a state set consisting of all possible initial states. The effect of each operation is uncertain. Conformant planning problem to solve is the initial state and the effect of action is uncertain circumstances, to find a certain plan to reach the target solution.

Flexible planning problem is a triad , in which is the initial state set, the set of operations, the target state set. Here, each operation for a specific action with a different degree of satisfaction. Some researchers define it as a four-tuple, Contains a measure of satisfaction. Flexible planning problem is to solve is the satisfaction of different actions (or preferences) measure, choose one of the most satisfactory or satisfactory enough efficient planning solution.

The above two kinds of non-classical planning problems, as the expansion of the classical planning problem from the uncertainty of the initial state and the action effect, are more in line with the needs of practical problems and suit the cognitive needs more consistently .

1.3 Convergence transformation planning method and fusion self-adaptive reasoning planning method

The convergent-transform programming method transforms the planning problem into other types of knowledge representations and is solved accordingly. The planning problem of the fusion logic representation only relies on the logical deductive method to be solved before 1992. Logical deductions need to be implemented under conditions that ensure reliability and completeness, which is not efficient. In 1992, Neumann [10] proposed the transformation of the planning problem to the propositional satisfiablility (referred to as SAT) to solve the problem of fusion to meet the satisfaction of the planning method. Ever since then, there have been many researches which have attempted to convert planning problem into model detection problem and linear programming problem. The corresponding planning systems Alt [2], MIPS [3], SGPlan [13] have displayed good performances.

In the narrow sense, it relies on the logical semantics of the planning problem itself, transforming it into some theorem proving problem of some logical representation, which calls for the corresponding adaptive reasoning programming method. Logic theorem proves that the system is to be solved, and the solution of the transformed problem can be reduced to the original solution to the original planning problem. The generally- accepted explanation is: logical deduction and reasoning technology are used, directly or indirectly, to verify the planning process; the former focuses on the transformation of knowledge representation, that is, the coding process, while the latter focuses on the use of reasoning technique to prove the process.

The transformation of planning problems to other types of knowledge representation is called encoding, and the new knowledge representation is called encode. Coding may be a specific kind of storage structure, relational structure, clause set, axiom or axioms, etc. For the convenience of description, the coding here only takes into account the representation of the axiomatic hierarchy and has not yet been converted into a conjunctive normal form (CNF) or a set of clauses.

Definition 2. (coding of the planning problem). For a particular coding scheme, the coding of a programming problem is a knowledge representation that can be characterized, either explicitly or implicitly, with respect to a given programming problem: (1) Prerequisites of operation (the coding of a planning problem) , the intrinsic relationship between effects; (2) the relationship between the operation; (3) the initial state and target conditions; 4) (optional) on the action and state constraints.

Definition 3. (coding combination of planning problems) For a particular coding scheme, the coding combination of planning problems is a knowledge representation that can explicitly or implicitly describe the following characteristics about a given planning problem: (1) the relationship between the operations; (2) the initial state and target conditions; (3) (optional) on the action and state constraints.

The coding combination is a kind of coding scheme which is converted into a logical representation. Coding and coding combination are two different concepts, with the former being the planning problem, which corresponds to a suitable overall knowledge representation, and does not distinguish between coding and coding combinations without causing confusion. Cresswell [4] transformed the planning problem into model detection problem.

2 Flexible programming method based on fusion fuzzy description logic

The flexible planning problem (also known as flexible programming problem) introduced introduces a measure of satisfaction to the operation in the planning problem, which is a relaxation of the classical programming. The flexible graph plan system is also the first planning system which addresses such problems.

Since the description of such problems has been difficult to unify, it is necessary to impose a fuzzy characterization on the operation, or a measure of the state preference, which is worth considering and measuring. The description logic ALC* was extended, and a fusion of fuzzy description logic (fuzzy description logic), fuzzy description of the state of the fuzzy description logic, FALC* (fuzzy ALC*) is proposed, which uses FALC* to express the state attributes, and the actions with the relation, which makes the attributes closer to the reality and the actions unique in preference. In addition, the problem definition is transformed into the inclusive relation of the fuzzy concept, and the action definition is transformed into the implicating relation of the fuzzy concept.

2.1 Fuzzy description logic problem representation

According to Badea's design method of fusion description logic frame, FALC* is used to code the planning problem as follows:

According to the standard semantics of fuzzy explanation I, it is interpreted as mapping $C(s)$ a truth value between $[0, 1]$ them. Then we call the subjective C truth degree of the attribute in the state to be n; $R(s, s')$ mapped to a truth value between $[0, 1]$, and then we call it from state s to state s', this action has preference l.

With the implication of fuzzy concept to express the dynamic axiom $DA : \langle C, n\rangle \to \exists\langle R, l\rangle.\{s'\} \wedge \forall\langle R, l\rangle.\langle D, m\rangle$ which, C and D represent that the concept in FALC *, R is FALC * in the relationship. The formula represents: the state s to meet the premise C, and the truth degree of C is $C(s)$, trigger preferences $R(s, s')$ for the action, so that the state s transferred to another state s', and the state s' to meet the true degree $D(s')$ of the property D.

For the knowledge base that needs to be described, the following type descriptions are included:

$$Pkg_2 < pkg \quad \text{etc;}$$

status description:

$$\begin{aligned}&ConnectCitywithMainR < MainRoad \wedge \exists\, forwardCity.\\&T \wedge \exists\, nextCity.\\&T \wedge \forall\, forwardCity.\\&City \wedge \exists\, nextCity.\\&City;\end{aligned}$$

action description:

$$\begin{aligned}&atTruckCity \wedge atPkgCity \wedge \neg onGuardTruck \wedge \langle ValuableP, k_2\rangle \to \exists\,(LoadTruck_3, l_2).\\&\{x(\neg atPkgCity \wedge onPkgTruck)\} \wedge \forall\,(LoadTruck_3, l_2).\\&(\neg atPkgCity \wedge onPkgTruck)\\&\text{coding etc.}\end{aligned}$$

2.2 Reasoning ability of fuzzy description logic

The post-transformation theory needs to assert the goal of deriving satisfaction degree n from the fuzzy knowledge base,$\sum | =\langle goal, n\rangle, i.e.$

$$\{\Gamma_S \cup \Gamma_D, \langle S(init), n\rangle\}| = ((\exists\,\langle plan, l\rangle.\{xgoal\}) \wedge (\forall\,\langle Plan, l\rangle.\langle goal, m\rangle))\,(init) \tag{3}$$

Among them, Γ_S is the static axiom set, it Γ_D is the dynamic axiom set; $init$ for the initial state, it $S(init), n\rangle$ means satisfying the attribute S, and its subjective truth degree is n .$(\exists\langle plan, l\rangle.\{xgoal\}) \wedge (\forall\langle Plan, l\rangle.\langle goal, m\rangle)$, indicating that there is a satisfaction degree l satisfying the target state, and satisfying the satisfaction degree m.

Since the pruning strategy is not adopted, the FPExp algorithm designed to find the optimal solution satisfies the requirements, but it needs to consider the appropriate pruning strategy to compress the search scale.

For planning solutions with satisfaction below a given limit, even if the length of the solution is very small, it is not considered as an effective solution. For the trade-off between length and satisfaction, the length of the solution is given priority.

2.3 Flexible programming method based on fusion fuzzy description logic

We design a flexible programming method which integrates fuzzy descriptive logic and solve the flexible programming problem by transforming the flexible programming problem into the knowledge representation problem of fuzzy description logic. This is a new attempt which corresponds to the complex planning problem to the non-classical logic problem, and use the inference technology of non-classical logic to deal with the transformed problem. Despite the far-from-expectation result , there is much to be developed with this method, such as adding pruning strategies and other heuristic search strategies.

3 Experiment and analysis

Currently, the enterprise decision-making method for the media enterprise data run features are not fully considered which led to no data on the steady-state decision to determine the enterprise decision-making. In this case, there is not enough clarity of case partitioning, resulting in poor comparability between the final information mining result and the actual data. In this paper, combined with the main features of the production mode of the media enterprise, the steady state judgment is carried out on the data. On this basis, a classification model is constructed to ensure the accuracy of the classification results and evaluate the effectiveness of the algorithm.

3.1 Experimental data

In this study, a media enterprise reasoning data was used, and the data of the first half of 2012 was selected as the research object. There are 427 attributes of the integrated data, besides the set key number and the data retention time automatically generated by the system, there are 425 condition attributes in total, all of which maintain the numerical characteristics. The steady-state, unsteady-state annotation of the operating data is obtained by referring to the steady-state operating parameters, and the decision-making attributes are obtained. And in order to better test the algorithm 3 (GARIv) and maintain the original data of each system unchanged, the original database of the same decision attributes is added.

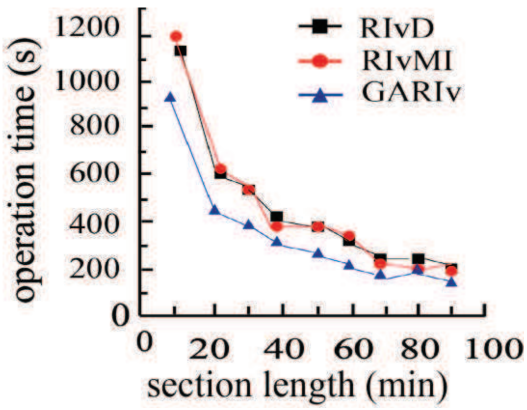
In addition, in order to evaluate the performance of the algorithm synthetically, a variety of partitioning methods are employed to design the data interval. If there is a decision class corresponding to a certain interval in the interval division, the data belonging to the same decision class are divided; it is classified into the same small interval, starting from the next interval starting from the different decision classes in turn.

The experiment is implemented on the workstation Intel Xeon(R) Processor (Four Core, 2.5GHz, 16GRAM). The algorithm is mainly realized by JAVA writing. In the calculation of the GARIv algorithm, two virtual machines are constructed to form two sites. In the experiment, the accuracy of classification can be confirmed, which can effectively guarantee the fairness of the experiment.

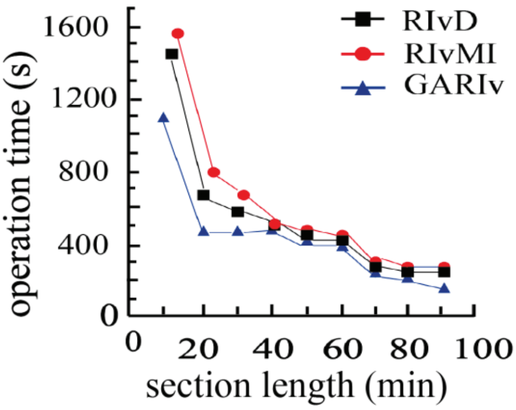
3.2 Experimental comparison

First, the experiment data of algorithm 1, algorithm 2 and algorithm 3 are combined with the selected experimental data. The time of the algorithm is determined by the interval length of the algorithm. Fig.1 (a) to Fig.1 (d) show $\lambda = 0.7$, the running time chart when the interval length is 10minutes, 20minutes, ..., 90minutes, respectively, when the number of attributes is different.

Fig. (a) To Fig. 2 (b), represents when $\lambda = 0.5, 0.7$ the running time required for selecting two attributes under different section lengths.



(a) Select 1 attribute



(b) Select 2 attributes

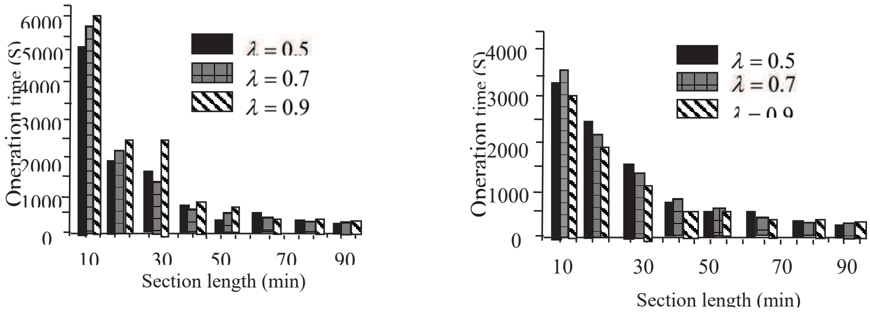
Figure 1: $\lambda = 0.7$, the running time of different decision algorithms

As can be seen from Figure 1:

- the length of the interval is increasing, the data object will be reduced by a factor of two, and the time required for the algorithm to run will decrease;

- When the length of the interval increases, the coincidence degree between the intervals will increase, which will increase the number of $\lambda-$ compatible class elements. If an attribute is added, the computation amount will increase. Therefore, the running time of the algorithm shows a nonlinear trend. The number of objects decreases, but the running time does not decrease with the interval length, but it shows an increasing state. Analysis of this situation may be due to it that although the length of the interval is increasing, but based on the number of $\lambda-$ compatible elements in an increasing number of state, thereby extending the running time;

- Compared with the algorithm RIvD and algorithm RIvMI, the two maintain relatively similar running times for a relatively small number of attributes, but the similarity decreases with the increase of the number of attributes, and the running time of the former will be longer. The reason is that the number of compatible classes increases with the increase of the number of attributes. However, it is necessary to judge whether the compatibility class is classified as positive domain when calculating the domain. It leads the algorithm RIvD to take longer time for the calculation, and for this algorithm RIvMI, the number of compatible class will increase with the property; however, its new conditional entropy, the combination of the original conditional entropy expansion can be achieved, and, therefore, the latter requires a shorter time.



(a) RIvD Select 3 attribute (b) GARIv Select 3 attributes

Figure 2: the running time required of the two algorithms when λ is different

Analysis of Figure 1, the required minimum run time is the GARIv algorithm, which is, theroretically, the algorithm running time for the basic half of the algorithm RIvMI. However, because the compatibility class is calculated, it is related to the transmission of partially compatible classes, and overall, the configuration of the virtual machine is difficult to match with the workstation, so that the actual time required for the algorithm GARIv exceeds its theoretical time; however, in terms of overall time, the algorithm is relatively low, and with the increase in the number of sites, this time advantage will be more prominent.

As shown in Fig.2, because the λ values are more irregular and more consistent, λ increases continuously. Their compatible classes are more refined and the number of compatible elements decreases. For the algorithm GARIv, the amount of compatible class traffic that needs to be done is larger, which results in a longer run time.

Experimental average classification accuracy and the traditional average classification accuracy were compared to evaluate the effectiveness of the three algorithms. The results are as follows:

Table 1: The average accuracies of classification

algorithm		Section length (unit:minutes)								
		10	20	30	40	50	60	70	80	90
RIvD	$\lambda = 0.5$	78.8%	80.3%	81.4%	80.2%	80.4%	80.3%	71.2%	70.9%	69.5%
	$\lambda = 0.7$	78.8%	84.4%	81.3%	82.4%	81.2%	79.4%	76.3%	70.5%	67.3%
	$\lambda = 0.9$	80.1%	81.5%	85.0%	81.9%	78.5%	79.3%	70.3%	66.9%	69.9%
RIvMI	$\lambda = 0.5$	85.2%	84.6%	86.6%	85.6%	82.6%	88.3%	79.6%	72.6%	67.9%
	$\lambda = 0.7$	84.5%	94.6%	89.6%	90.3%	92.3%	92.6%	79.2%	73.6%	62.6%
	$\lambda = 0.9$	83.5%	90.2%	88.3%	92.6%	88.3%	92.3%	84.2%	79.3%	72.5%
GARIv	$\lambda = 0.5$	77.3%	91.6%	86.3%	89.6%	86.3%	86.3%	80.3%	72.3%	67.5%
	$\lambda = 0.7$	81.2%	93.4%	84.3%	82.5%	85.5%	92.2%	82.5%	81.0%	72.9%
	$\lambda = 0.9$	78.2%	91.1%	88.3%	83.5%	82.7%	84.5%	75.2%	80.5%	71.2%

After analysis of the table above, in many traditional analysis methods, the classification effect of kNN is the best; and RBF, REPTree have a relatively low accuracy, because the multi-source information in the steady-state judgment calls for a certain interval of data to judge, which is difficult to achieve through only one stream of data. The reason why the kNN accuracy in the classification is better is that it takes the data value of the same sub-section for the same paragraph of its decision-making class. Thus, when a neighbor k is computed, the corresponding decision class is as likely as the adjacent test result, and if the value k keeps increasing, a cross-class phenomenon is most likely to occur. This approach effectively reduces the information dimension and enhances the accuracy and efficiency of multi-source information decision making.

The three algorithms proposed in this paper are implemented on the basis of interval division, and are easier to satisfy the requirements of the decision-making stability, and thus have better accuracy than the traditional methods.

4 Conclusion and outlook

With the demand of problem representation, the programming method of the deductive logic and reasoning technology has been paid more and more attention to by the researchers of the related fields. A variety of non-classical logic calls for the need of the fusion knowledge representation, adding the description of modal, time and ambiguity etc. Many researchers use various techniques such as satisfiability judgment technique, first-order predicate lifting technique, fuzzy descriptive logic to express satisfaction degree of planning, and fusion modal logic representation and deduction reasoning technology to deal with various complicated planning problems, which has achieved a recognized outstanding recognition, promoting the intelligent decision support and self-adaptive reasoning of mutual promotion and integration. The uniqueness of media enterprises does not allow the one-sided decision-making system to fully meet the needs of its development, which means that in the future research the features of media enterprises should be taken into consideration so that an exclusive decision-making system can be devised and developed to meet the requirements of this industry in terms of operation and management.

Bibliography

[1] Aleksander I., Farreny H., Ghallab M. (1987); *Robot technology: decision and intelligence*, Hardcover, 1987.

[2] Carmen R., Defraeye M., Nieuwenhuyse I.V. (2015); A decision support system for capacity planning in emergency departments, *International Journal of Simulation Modelling*, 14(2), 299-312, 2015.

[3] Cresswell S.N., Mccluskey T.L., West M.M. (2013); Acquiring planning domain models using locm, *Knowledge Engineering Review*, 28(2), 195-213, 2013.

[4] Fox M., Long D. (2003); Pddl2.1: an extension to pddl for expressing temporal planning domains, *Journal of Artificial Intelligence Research*, 20(20), 61-124, 2003.

[5] Galichon A., Henry-Labordere P., Touzi N. (2014); A stochastic control approach to no-arbitrage bounds given marginals, with an application to lookback options, *Annals of Applied Probability An Official Journal of the Institute of Mathematical Statistics*, 24(1), 312-336, 2014.

[6] Ghallab M., Nau D., Traverso P. (2014); The actor's view of automated planning and acting: a position paper, *Artificial Intelligence*, 208(1), 1-17, 2014.

[7] Laborie P. (2003); Algorithms for propagating resource constraints in ai planning and scheduling: existing approaches and new results, *Artificial Intelligence*, 143(2), 151-188, 2003.

[8] Mahmood N. (2015); Connectionism strategy for industrial accident-oriented emergency decision-making: a simulation study based on pcs model, *International Journal of Simulation Modelling*, 14(4), 633-646, 2015.

[9] Mali A.D., Tang M. (2016); Path-cost bounds for parameterized centralized variants of a* for static and certain environments. *International Journal on Artificial Intelligence Tools*, 25(04), 1650028, 2016.

[10] Neumann B. (ed.) (1992); *Proceedings of the 10th European conference on Artificial intelligence. European Conference on Artificial Intelligence*, John Wiley & Sons, Inc, 1992.

[11] Sharon G., Stern R., Goldenberg M., Felner A. (2013); The increasing cost tree search for optimal multi-agent pathfinding, *Artificial Intelligence*, 195, 470-495, 2013.

[12] Shivaie M., Ameli M.T. (2014); *An implementation of improved harmony search algorithm for scenario-based transmission expansion planning*, Springer-Verlag, 2014.

[13] Wah B. W., Chen Y. (2006); Constraint partitioning in penalty formulations for solving temporal planning problems, *Artificial Intelligence*, 170(3), 187-231, 2006.

[14] Zhang Z.H., Hu C. (2015); Multi-model stability control method of underactuated biped robots based on imbalance degrees, *International Journal of Simulation Modelling*, 14(4), 647-657, 2015.