Behavior-based decision making: a tutorial

Mohammad Samadi Gharajeh

International Journal of Dynamics and Control

ISSN 2195-268X
Volume 6
Number 4

Int. J. Dynam. Control (2018)
6:1816-1840
DOI 10.1007/s40435-018-0412-4

International Journal of Dynamics and Control (2018) 6:1816–1840
https://doi.org/10.1007/s40435-018-0412-4

CrossMark

Behavior-based decision making: a tutorial

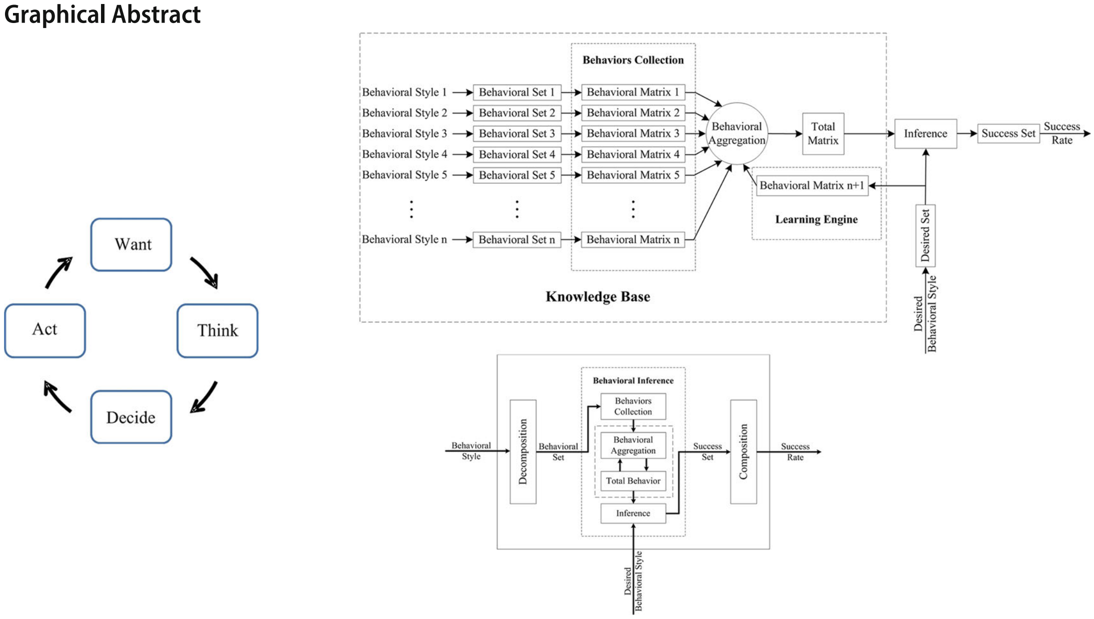
Mohammad Samadi Gharajeh[1]

Received: 6 February 2018 / Revised: 16 February 2018 / Accepted: 19 February 2018 / Published online: 8 March 2018

Abstract

There are ingenious characteristics in humanistic behaviors so that they can be utilized by the most developers to design smart and complex systems. This paper proposes a novel, knowledge and learning based method called behavior-based decision making, BBDM, in control and system engineering. It is an expert decision support system containing the learning ability to work based on humanistic behavioral reasoning. The knowledge base is built by the system based on various behavioral styles (e.g., safe) associated to other systems and humans. BBDM uses the knowledge-based information to make appropriate decisions when any desired behavioral style is requested from the system. It specifies a success rate for any desired style based on the obtained knowledge base with the aid of a behavioral inference system. This procedure can be used to select a proper system or human to accomplish a requested job. All operations of the BBDM method are performed by a proposed behavioral decision system, called BDS, which consists of three main units: decomposition, behavioral inference, and composition. The decomposition unit splits any behavioral style into several optional features (e.g., safety). The behavioral aggregation sub-unit aggregates all behavioral styles obtained by the system to define the total behavior. The behavioral inference unit produces a success set for any desired behavioral style. Finally, the composition unit converts success set to success rate to specify the success probability of the desired style. Simulation results show that the proposed method has a high efficiency compared to some of the existing decision-making methods.

Graphical Abstract



Keywords Decision system · Knowledge-based system · Learning ability · Humanistic behavior · Behavioral inference

✉ Mohammad Samadi Gharajeh
mhm.samadi@gmail.com; m.samadi@iaut.ac.ir;
m.samadi@ieee.org

[1] Young Researchers and Elite Club, Tabriz Branch, Islamic Azad University, Tabriz, Iran

List of symbols

F	Feature collection
f	Any feature in the feature collection
B	Behavioral set
η	The importance factor in the behavioral set
∂	The belongingness factor in the behavioral set
ϕ	The number of decimal places in the identical importance function
d_x	The degree of any feature in the belongingness functions
d_s	The degree of the sensible feature in the belongingness functions
v_s	The value of the sensible feature in the belongingness functions
α	The relative factor in the inverse belongingness function
β	The relative factor in the relativism belongingness function
T	Total matrix
γ	Trust amount in the behavioral matrix
D	Desired set
S	Success set
σ	The difference amount in the inference functions
φ	The impact rate of attribute 'truth' in the inference functions
ω	The impact rate of attribute 'growth' in the inference functions
R	Success rate

1 Introduction

Intelligent methods [1,2] play a major role in the development process of smart systems. They use some of the brilliant properties related to humans and animals to develop complex and smart systems. Because various systems such as mechanical systems [3], electrical systems [4], and computing systems [5] are applied in most of the modern applications, they can be developed by intelligent methods. To achieve this goal, some of the well-known methods such as artificial neural networks (ANNs) [6,7], machine learning [8,9], multi-criteria decision making (MCDM) [10,11], fuzzy logic [12–15], evolutionary computing [16,17], learning theory [18,19], and deep learning in neural networks [20] are used in such applications in the last decades.

Smart systems [21,22] involve complex and sophisticated structures so that they use intelligent procedures to make appropriate decisions under various conditions. Because humanistic reasoning solves a large number of computation problems, they can be utilized by smart systems to considerably enhance the system throughput. Though various intelligent methods are presented by researchers whole the world, but most of them work based on pre-defined information instead of humanistic behaviors. Besides, they provide static decision making, in the most cases, without having potential humanistic characteristics. There are some unpredicted conditions that can be solve by special humanistic techniques. Whereas the existing methods are not completely applicable for all environmental conditions, a novel decision-based method is essential now to make appropriate decisions based on humanistic behaviors.

A knowledge-based and auto-learning method is proposed in this paper that works based on various humanistic behaviors [23]. It can be used in most of the smart systems which interact with other systems and humans. The proposed system utilizes some brilliant properties of the humanistic behaviors to make appropriate decisions under various environmental conditions. Humanistic reasoning is considered by the proposed decision system to achieve high efficiency in face of certain and unforeseen situations. This system collects some prior behavioral styles associated to other systems and people in order to make proper decisions when any desired behavioral style is requested from the system. A success rate is specified for the desired style based on the obtained behaviors with the aid of a behavioral inference engine. Results of the desired behavioral styles can be obtained by the knowledge base to enhance the learning ability of the system. It is worth noting that the main goal of the proposed decision method is to present a novel humanistic behavioral technique to science world to solve some of the existing control problems.

The rest of this paper is organized as follows. Some of the existing intelligent methods are presented in Sect. 2. The basic characteristics of the humanistic behaviors are explained in Sect. 3. The proposed decision method is described in Sect. 4 to represent various aspects of the proposed behavior-based system. This section, moreover, describes the main units of this system including decomposition, aggregation, inference, and composition. Section 5 explains the proposed method with a practical example. Section 6 discusses on the analysis results of the proposed system compared to existing intelligent methods. The proposed method is compared to some of the existing knowledge-based methods in Sect. 7. Finally, the paper is concluded by Sect. 8.

2 Background

ANNs [6,7] work based on biological neuronal systems that are presented by McCalloch and Pits in 1943. The authors presented a novel and potential non-algorithmic method for processing the different information. The architecture of ANNs is similar to a human's brain so that learning algorithms can use it to design intelligent systems. The algorithms of ANNs are categorized into three main groups as supervised, reinforced, and unsupervised learning algorithms. The lack of potential dynamic decision making is the major limitation of these networks.

Machine learning [8,9] is an artificial intelligent method that utilizes some of the humanistic learning abilities for appropriate decision making. It constructs a general and smart model based on humanistic thinking. The model involves an accurate and comprehensive architecture to predict some of the upcoming conditions by learning information. That is, it should have a general architecture to be applicable not only for predicted situations but for un-predicted conditions. Machine learning is categorized into two groups as constructing an informatics model and providing a systematic structure.

MCDM [10,11] makes proper decisions in the presence of multiple criteria. For example, routing packets from the source station to the destination station in computer networks can be performed based on multiple criteria such as traffic, distance, and energy. MCDM can be commonly used in control and engineering systems. The development process of MCDM is closely related to the advancement of computer technology. That is, the rapid progress in computer technology leads to systematically analyze complex MCDM problems. In general, the finite number of alternative solutions and the infinite number of alternative solutions are two basic types of MCDM problems.

Fuzzy logic [12–15] and fuzzy sets [24] are potential tools for appropriate decision making under un-predicted conditions. They introduced by Lotfi-Zadeh in the mid-1960s. Fuzzy logic applies some linguistic terms (e.g., cold and warm) to construct complex systems based on humanistic reasoning. It uses the main concepts of fuzzy sets instead of classic sets. Each element of fuzzy sets takes a degree of belongingness in the range of [0, 1]. In contrast, each element of classic sets only takes two logic values: 'true' and 'false'. A fuzzy system can make an appropriate decision in the presence of uncertain and ambiguous information. Therefore, fuzzy logic is a powerful tool for modeling the complex systems based on human-based experiences.

Evolutionary computing [16,17] is a biological method that consists of natural features including reproduction, mutation, and survival. It operates similar to a natural evolution in a way that the population of possible solutions must be better from one generation to another generation. Evolutionary techniques are useful to design those of the sophisticated applications which need to much more search spaces. In fact, they work similar to stochastic search techniques that appear different states from one generation to another. Having a high complexity structure is the remarkable weakness of the evolutionary computing.

Learning theories [18,19] are composed of some learning techniques that describe how a human can learn. Some of the current technologies such as multimedia, communication, and information can be developed by such theories. In general, learning theory is categorized into two basic groups: behaviorism and constructivism. Behaviorism theory is presented based on a teaching procedure that is known as a self-evident and objective theory. It briefly concentrates on those of the environmental conditions which are related to humanistic natural behaviors. In contrast, constructivism theory is known as a unique learning theory that can be involved with perceptional psychology. It is presented based on a learner's ability in general learning mechanisms. The cognitive and social methods are two basic types of constructivism theory.

Deep learning in neural networks [20] is a potential set of techniques for learning purposes. It offers proper solutions to many problems in some of the applications such as speech recognition, natural language processing, and image recognition. These networks are distinguished from the popular single-hidden-layer neural networks by their depth. Each layer of neural nodes learns on a distinct set of multiple features based on the output of the previous layer. It aggregates and recombines the features received from the previous layer. All of these significant characteristics lead deep learning to be used in various areas in order to solve some of the existing sophisticated problems.

Bayesian network [25,26] is an annotated directed graph, which encodes some probabilistic relationships between distinctions of interest under uncertain conditions. It uses a human-oriented qualitative structure to facilitate relationships between a system and a user incorporating the probabilistic model. Artificial intelligence researchers have used these networks to encode the expert knowledge of intelligent applications. Bayesian models can be applied for a wide range of applications such as prediction, anomaly detection, diagnostics, and automated insight.

Adaptive neuro-fuzzy inference system (ANFIS) [27,28] is a data mining methodology that works based on a combination of fuzzy logic and neural networks. It determines various clustering values in fuzzy sets, predicts membership functions during the training process, and calculates the weights by a neural network model. ANFIS architecture is defined as an adaptive network, which applies supervised learning on learning algorithm. It works similar to the model of Takagi–Sugeno fuzzy inference system by using the five pre-defined layers.

The above intelligent methods have some remarkable limitations as following:

- Most of them make appropriate decisions based on some static conditions according to the pre-defined structures (e.g., a graph). Thus, they cannot be carefully applicable under all un-predicted conditions.

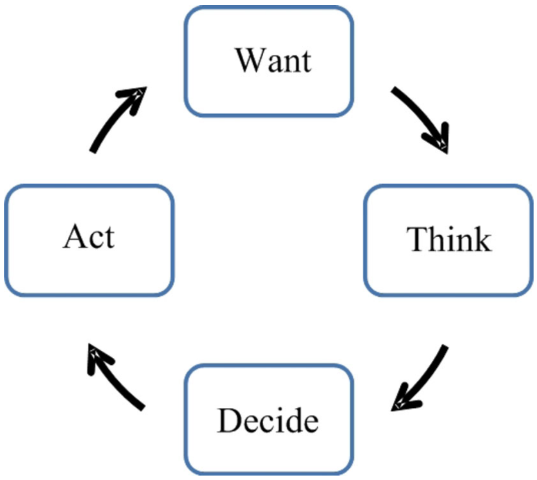


Fig. 1 A schematic of the humanistic behavior cycle

- Most of them are offered to make proper decisions based on systematic operations instead of humanistic behaviors.
- Most of them are system-oriented methods instead of being behavior-oriented techniques. Behavior-oriented techniques involve some of the genius properties that cannot be found in system-oriented methods. Therefore, they are not able to determine proper decisions in most of the humanistic environments.
- The humanistic behavior-based reasoning is different with the humanistic data-based reasoning. The lack of a real behavioral reasoning in the existing intelligent methods causes some problems to be emerged on some un-predefined conditions.

3 A glance on humanistic behaviors

A human interacts with other people through some humanistic behaviors [23,29,30]. He makes appropriate decisions under various environmental conditions based on the behavioral cycle shown in Fig. 1. The humanistic behavior cycle includes four basic elements: want, think, decide, and act. When a human decides to do a work or job, he is placed in the wanting phase. In the thinking phase, he thinks about other people and things based on the knowledge base of prior information. After analyzing most of the available information, he makes appropriate decisions related to the requested job. Finally, he acts accordance with the adopted decision.

Any humanistic behavior is composed of various features to describe the behavior, clearly. For example, a driving behavior can be indicated by some features such as safety, accuracy, and speed. The importance weight of the features can be the same or differ with each other. A driver can pay attention to the safety feature more than other features. Any behavior is indicated by some styles called behavioral style. For example, the driving behavior can be indicated by some behavioral styles such as safe, accurate, and fast. When a person drives accurately, he pays attention to the accuracy features more than other features.

Finding a success rate for any desired behavioral style (e.g., the safe driving) is one of the important decisions in humanistic interactions. Fig. 2 describes an example to determine a success rate for any desired driving behavior based on some prior experiences. Fig. 2a represents the knowledge base associated to a driver's driving records. Each entry indicates one of the driving instances, which is accomplished by the driver anywhere and anytime. The instances are presented by various behavioral styles such as safe, accurate, etc. Fig. 2b represents the success rates estimated for the desired behavioral styles. These rates are inferred based on all the instances of the driving behavior collected in the knowledge base. For example, when a driver having the safe driving style is needed, it is estimated that the current driver can accomplish the job with the success probability of 0.4. Any success rate specifies the success probability of the driver that is equal to a value in the range of [0…1]. Note that it can be estimated for any desired behavioral style by humanistic reasoning without being the desired style in the knowledge base. All of the above statements can be performed by humanistic behavioral reasoning. This paper proposes a new decision making system based on humanistic behavioral reasoning similar to the above statements. The main goal of the proposed system is to solve existing problems or to facilitate current applications in control and system engineering.

4 Behavior-based decision making

Human is the greatest creature in the world. It has the potential capabilities more than other creatures in the universe. Therefore, humanistic behaviors can be used in control and system engineering to solve some of the existing problems or abolish some of the existing limitations. A novel, knowledge and learning based decision system called behavior-based decision making, BBDM, is proposed in this paper that uses humanistic behavioral reasoning in control and system engineering. The main goal of BBDM is to specify a success rate for any desired behavioral style based on the knowledge base of prior behavioral styles. This method can be used to select a human or a smart system having the highest success rate from the list of various humans and systems in order to accomplish a requested job. It tries to work similar to humanistic behavioral thinking in face of different conditions. The proposed system is described in the subsequent sub-sections.

Figure 3 shows a schematic to describe an overall view and the main goal of the proposed decision making system. Any behavioral style (e.g., safe, accurate, and fast) is presented by a behavioral set. Afterward, behavioral set is converted to a behavioral matrix by the behavioral aggregation unit in order to produce a total matrix based on the obtained knowledge base. Note that total matrix indicates total behavior of the system. When a desired behavioral style is requested from the system, it is converted to a desired set by behavioral inference process to produce a success set for the desired style. Finally, the success set is converted to a success rate. This rate can be varied based on the desired style and the obtained knowledge base. It is not necessary that the desired behavioral style is existed in the knowledge base because the success rate can be inferred based on humanistic behavioral reasoning with the aid of behaviors collection. After the success rate is calculated by the system and the desired behavior is accomplished by anyone (e.g., a driver in the driving behavior), the results

Knowledge base

#	***Behavioral style***
1	Safe
2	Safe & Accurate
3	Slow
4	Fast & Safe
5	Safe & Slow
6	Dangerous & Fast
7	Fast
8	Dangerous
9	Fast & Accurate
10	Slow
...	...
n	...

(a)

Inference

Desired behavioral style	***Success rate***
Safe	0.4
Accurate	0.2
Slow	0.3
Fast	0.4
Reliable	0.5
...	...

(b)

Fig. 2 Determining a success rate for any desired driving behavior. **a** Various instances of the behavior, **b** specifying a success rate for each desired behavioral style

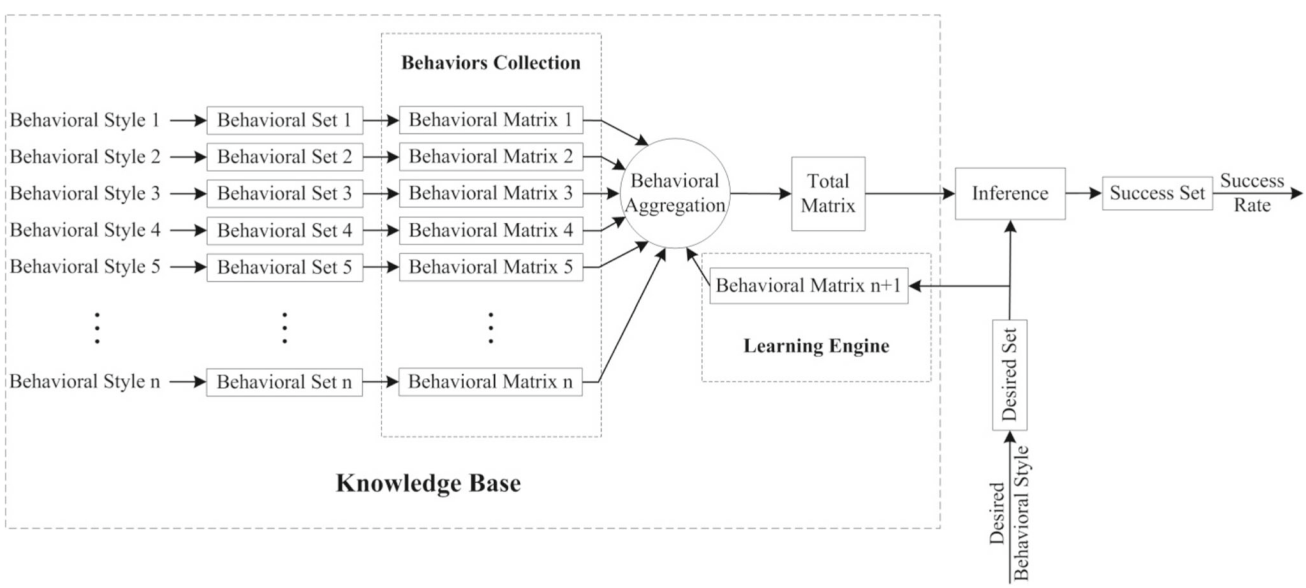


Fig. 3 An overall view of the proposed BBDM method

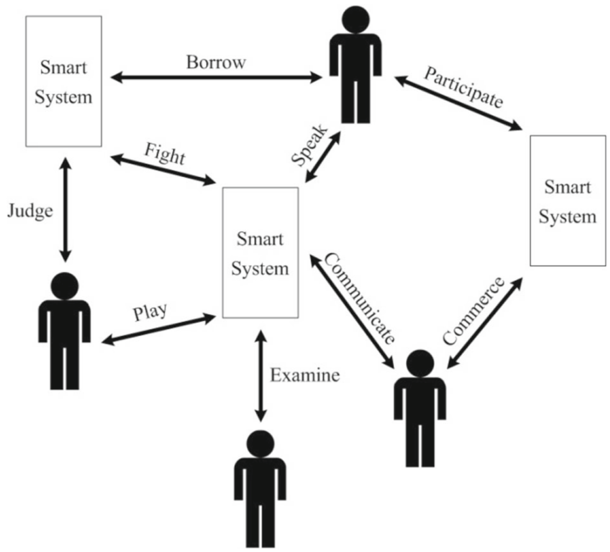


Fig. 4 A schematic of the network model in HBN

of the completed behavior can be entered to the system in the learning engine. Note that behavioral styles, behavioral sets, behavioral matrixes, and total matrix are the main elements of the knowledge base. All of the above units are explained in the next sub-sections.

4.1 Humanistic behavioral network

In this paper, smart systems and people interact with each other into a network model called humanistic behavioral network, HBN. They can utilize the proposed decision system to make appropriate decisions on different behaviors under various environmental conditions. Note that people can also use their natural reasoning in this network. Figure 4 shows a schematic of HBN included to some smart systems and people. As depicted in this schematic, any system communicates with others through different behaviors.

4.2 Behavior structure

In the proposed decision system, behavior structure is defined as follows. Various behavioral styles (e.g., safe) are stored in a collection called style collection, SC. This collection is one of the main components of the knowledge base that involves initial obtained behavioral styles. Any behavioral style and desired behavioral style is composed of several optional features (e.g., safety) to describe characteristics of the behavior. All features are placed in a collection called feature collection, F. In this collection, any feature has two attributes: degree and value. Both attributes take a value in the range of (0, 1] to represent their quantity amounts. Because any human pays attention to all features of the behaviors in the real world, it is better that both attributes are not equal to 0. Attribute 'degree' specifies effectiveness weight of any feature in the feature collection. It is called importance factor and denoted by symbol η. Sum of the degrees into any behavioral style must be equal to 1. For example, if 'safety', 'accuracy' and 'speed' are three features of a driving behavior so that it is supposed that 'safety' is more important than 'accuracy' and 'accuracy' is more important than 'speed', the importance factors of the features can be considered as 0.5, 0.3, and 0.2, respectively. Attribute 'value' determines quantity amount of any feature into any behavioral style. It is called belongingness factor and denoted by symbol ∂. If a driver drives the car vary safe and accurate in the driving behavior represented above, the belongingness factors of features 'safety', 'accuracy' and 'speed' can be considered as 1, 0.8, and 0.1, respectively. In fact, any behavioral style determines the value of any feature into behavioral style. Any behavioral style is represented by a set called behavioral set. This set consists of all attributes defined in the feature collection as follow

$$B = \left\{ \frac{(\eta_B(x), \partial_B(x))}{x} \mid x \in F \right\} \tag{1}$$

where x is feature name (e.g., 'safety', 'accuracy' and 'speed') in set F, $\eta_B(x)$ is importance factor of feature x, and $\partial_B(x)$ is belongingness factor of feature x. The names of the features are fixed in behavioral set B, but $\eta_B(x)$ and $\partial_B(x)$ can be changed for any behavioral style. Note that $\eta_B(x)$ is often specified before obtaining any behavioral style, but $\partial_B(x)$ is always determined when a behavioral style is obtained by the system.

A driving behavior is considered here in a way that the feature collection is defined as $F = \{\text{'safety'}, \text{'accuracy'}, \text{'speed'}\}$. If a behavioral style obtained by the system is very safe and accurate, its behavioral set can be specified as $B = \left\{ \frac{(0.5, 1.0)}{\text{safety}}, \frac{(0.3, 0.8)}{\text{accuracy}}, \frac{(0.2, 0.2)}{\text{speed}} \right\}$. Suppose that the importance of feature 'safety' is more than the importance of feature 'accuracy' and the importance of feature 'accuracy' is more than the importance of feature 'speed'; thus, the importance factors of the features are considered as 0.5, 0.3, and 0.2, respectively. Because the obtained behavioral style is considered as very safe and accurate, the belongingness factors of the features are considered as 1.0, 0.8, and 0.2, respectively. Figure 5 shows graphical representation of the features 'safety', 'accuracy' and 'speed' in the above behavioral style. The axis x represents the importance factors of the features and the axis y represents the belongingness factors of the features.

4.3 Behavioral decision system

All operations of the BBDM method are performed by a system called behavioral decision system, BDS, as shown

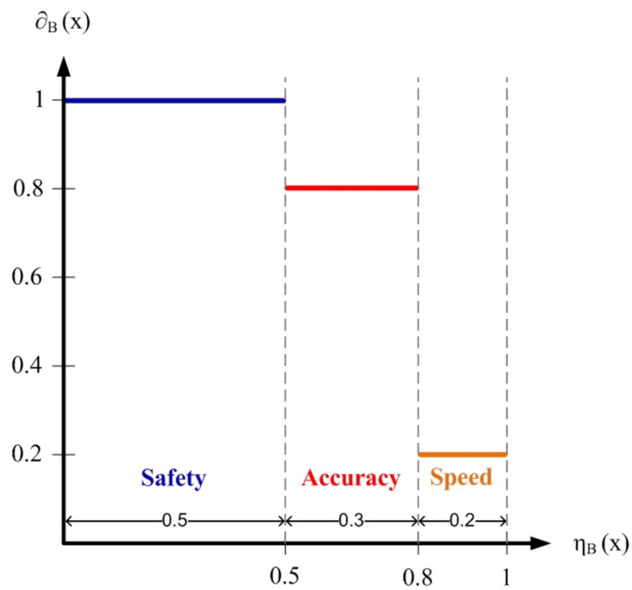


Fig. 5 Graphical representation of three features in a driving behavior

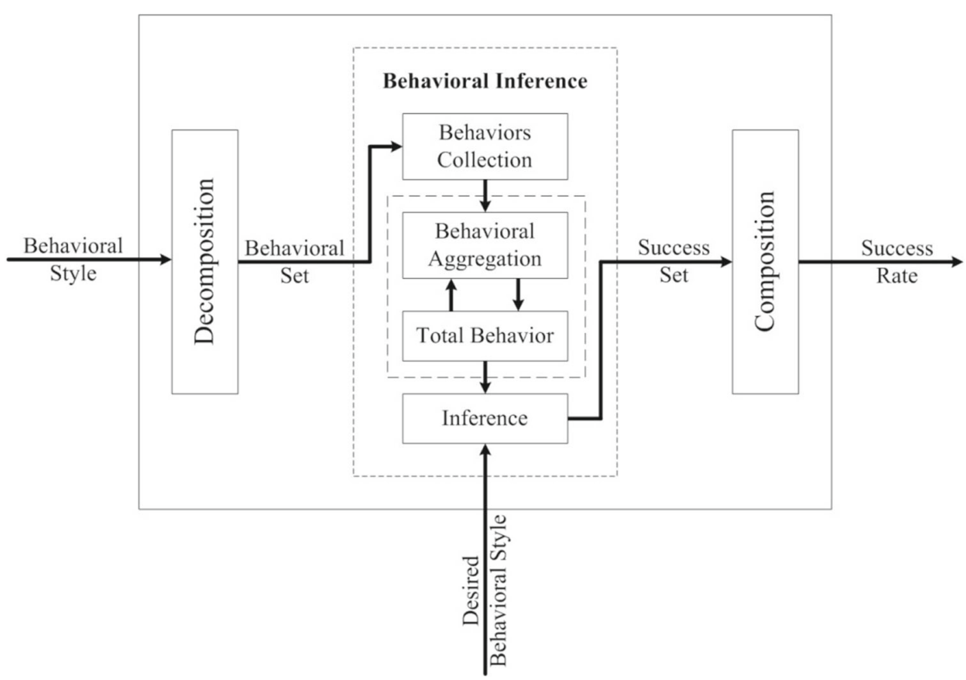
in Fig. 6. BDS consists of three main units: decomposition, behavioral inference, and composition. It stores various behavioral styles related to any behavior (e.g., the driving behavior) to produce a success rate based on the obtained knowledge base. When a behavioral style is obtained by the system, degrees and values of the features are determined by manual way or one of the proposed decomposition methods described in Sect. 4.3.1. All behavioral styles are aggregated together by one of the proposed aggregation methods to construct a total behavior as described in Sect. 4.3.2.3. When a desired behavioral style is requested from the system, BDS produces a success set based on the obtained knowledge base and the desired style as described in Sect. 4.3.2.4. Finally, success set is converted to a success rate by one of the proposed composition methods described in Sect. 4.3.3.

4.3.1 Decomposition

Decomposition unit is defined to convert any behavioral style to a behavioral set. Because both of importance factors and belongingness factors can be varied for any behavioral style, this unit is used when a behavioral style is obtained by BDS. Decomposition methods are proposed in this sub-section to determine importance and belongingness factors based on some of the humanistic thinking in face of different conditions. They are categorized into two groups: decomposer of importance factors and decomposer of belongingness factors.

4.3.1.1. Decomposer of importance factors Importance factors of the behavioral set can be specified by manual way or one of the proposed decomposer methods called decomposer of importance factors. They are often defined by manual way if features are absolutely specified based on a practical humanistic behavior. An overall view to determine importance factors of any behavioral set by the proposed decomposer of importance factors is shown in Fig. 7. Features are represented by the series of $\{f_1, f_2, \ldots, f_n\}$ and importance factors are represented by the series of $\{\eta_1, \eta_2, \ldots, \eta_n\}$ where n is the number of features in feature collection F. Importance factors are specified by three decomposer methods: identical importance function, incremental importance function, and decremental importance function.

Fig. 6 A schematic of BDS



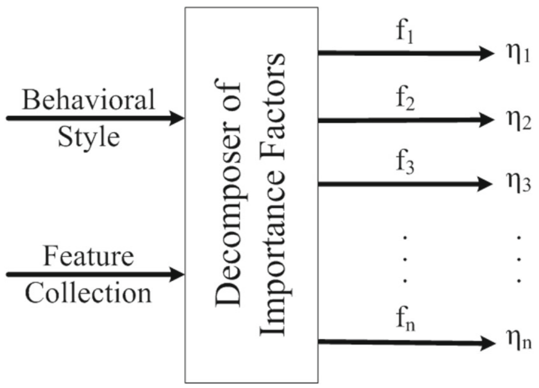


Fig. 7 A schematic for decomposer of importance factors

4.3.1.1.1. Identical importance function Effectiveness weights of all features are the same in some humanistic thinking. This type of thinking is performed in the proposed decision system by a decomposer method called identical importance function. This method determines degrees of the features in a way that all features have the same importance. The algebraic form of the identical importance function is defined as follow

$$\forall \mathrm{x} \in \mathrm{n}, \eta(x) = \begin{cases} \left\lfloor \frac{10^{\phi}}{n} \right\rfloor \cdot 10^{-\phi}, & x < n \\ 1 - \left((n-1) \cdot \left(\left\lfloor \frac{10^{\phi}}{n} \right\rfloor \cdot 10^{-\phi}\right)\right), & x = n \end{cases} \tag{2}$$

where x is identifier number of each feature in feature collection F, n is the number of features, and ϕ is the specified number of decimal places. For example, if there are four features in set F and ϕ is equal to 2, the importance factors of all features are equal to 0.25.

4.3.1.1.2. Incremental importance function Effectiveness weights of the features cannot be the same in some humanistic reasoning. In this thinking type, the degree of each feature is more than the degree of prior feature. This type is utilized in the proposed system by a decomposer method called incremental importance function. The algebraic form of this function is determined as follow

$$\forall \mathrm{x} \in \mathrm{n},\ \eta(x) = \begin{cases} \frac{2x}{n.(n+1)}, & x < n \\ 1 - \left(\sum_{y=1}^{y=n-1} \frac{2y}{n.(n+1)}\right), & x = n \end{cases} \tag{3}$$

where x indicates identifier number of each feature in feature collection F, y represents identifier number of each feature, and n indicates the number of features. For example, if there are four features in set F, the importance factors of them are calculated as {0.1, 0.2, 0.3, 0.4} .

4.3.1.1.3. Decremental importance function In some humanistic thinking, importance factors of the features can be valued from the high importance to the low importance. This type leads to importance factors have a decremental flow in a way that the degree of each feature is less than the degree of previous feature. It is applied in the proposed system by a method called decremental importance function. The algebraic form of this function is represented as follow

$$\forall \mathrm{x} \in \mathrm{n},\ \eta(x) = \begin{cases} \frac{2(n-x+1)}{n \cdot (n+1)}, & x < n \\ 1 - \left(\sum_{y=1}^{y=n-1} \frac{2(n-y+1)}{n \cdot (n+1)}\right), & x = n \end{cases} \tag{4}$$

where x is identifier number of each feature in feature collection F, y is identifier number of each feature, and n is the number of features. For example, if there are four features in set F, the importance factors of them are determined as { 0.4, 0.3, 0.2, 0.1} .

4.3.1.2. Decomposer of belongingness factors Although belongingness factors of the features are usually determined by manual way based on realistic results, but they can be determined regarding to various types of the humanistic thinking. In this case, belongingness factors are calculated by the proposed decomposition methods called decomposer of belongingness factors according to the value of a sensible feature. This is done using quantity amounts of the behavioral style, feature collection, and the sensible feature as shown in Fig. 8. Features are represented by the series of {f_1, f_2, …, f_n} and belongingness factors are represented by the series of {∂_1, ∂_2, …, ∂_n} where n is the number of features. Note that the value of the sensible feature is specified by manual way and the values of other features are calculated based on the value of the sensible feature. Decomposer of belongingness factors is categorized into two groups: inverse belongingness function and relativism belongingness function.

4.3.1.2.1. Inverse belongingness function There are various thinking types in humanistic reasoning to forecast some of the future events. The indirect or inverse thinking is one of such thinking types so that the most of events can be predicted based on the indirect humanistic reasoning. In this type, there is an inverse pre-defined relation between all features to spec-

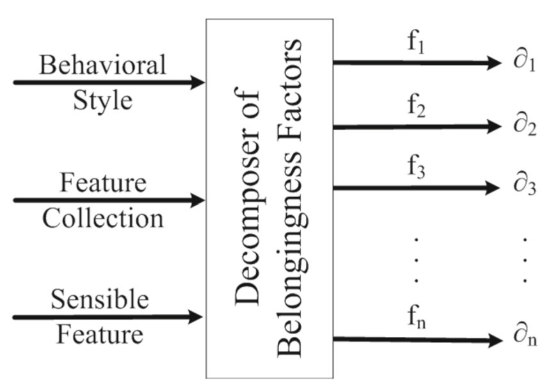


Fig. 8 A schematic for decomposer of belongingness factors

ify belongingness factors based on importance factors of the behavioral set. It is used in the proposed decision system by a decomposer method called inverse belongingness function. This method indicates that when a human focuses on a special feature, other features are neglected by him; and vice versa. If a driving behavior is considered here with three features including 'safety', 'accuracy' and 'speed' in a way that the safety is sensible feature, the following statements are some instances of the inverse thinking:

- If the driver pays more attention to safety, he pays less attention to other features.
- If the driver pays less attention to safety, he pays more attention to other features.
- If the driver pays too much attention to safety, he pays no attention to other features.
- If the driver pays no attention to safety, he pays too much attention to other features.

The above statements represent that if a human increasingly concentrates on a special feature, other features are neglected by him. The algebraic form of the inverse belongingness function is defined as

$$\forall \mathrm{x} \in \mathrm{n},\ \partial(x) = \begin{cases} \left| 1 - \frac{|d_s - d_x|}{v_s * d_x} \right|, & \frac{v_s * d_x}{d_s} > 1 \\ \left| \frac{v_s * d_x}{d_s} - \alpha \, |d_s - d_x| \right|, & 0.1 < \frac{v_s * d_x}{d_s} \leq 1 \\ 1 - \frac{v_s * d_x}{d_s}, & \frac{v_s * d_x}{d_s} \leq 0.1 \end{cases} \tag{5}$$

where x is identifier number of each feature in feature collection F, n is the number of features, d_x is the degree of feature x, d_s is the degree of sensible feature, v_s is the value of sensible feature, α is the relative effectiveness factor between sensible feature and other features. Note that α takes a value in the range of [0, 0.1]. The mathematical equation $\alpha \, |d_s - d_x|$ is used to balance quantity amounts between sensible feature and other features. If the belongingness factor is become greater than 1, it will be set equal to 1.

4.3.1.2.2. Relativism belongingness function The direct or accordance thinking is one of the humanistic thinking types so that the most of features can be determined by a special feature through a directed procedure. In this type, a human can specify belongingness factors of all features via a directed relation among the features. This process indicates that if a human pays attention to a special feature, he will also pays attention to other features. This type of humanistic thinking is considered in the proposed decision system by a proposed decomposition method called relativism belongingness function to determine belongingness factors of the features. If a driving behavior is assumed here with three features including 'safety', 'accuracy' and 'speed' in a way that the accuracy is sensible feature, the following statements are some instances of the direct thinking:

- If the driver pays more attention to accuracy, he pays more attention to other features.
- If the driver pays less attention to accuracy, he pays less attention to other features.
- If the driver pays too much attention to accuracy, he pays too much attention to other features.
- If the driver pays no attention to accuracy, he pays no attention to other features.

The above statements describe that if a human concentrates on a special feature, he also concentrates on other features. The algebraic form of the relativism belongingness function is determined as

$$\forall \mathrm{x} \in \mathrm{n},\ \partial(x) = \begin{cases} 1, & (v_s > 0.8) \,\&\, (d_x > d_s) \\ |v_s - \beta(d_s - d_x)|, & otherwise \end{cases} \tag{6}$$

where x is identifier number of each feature in feature collection F, n is the number of features, d_x is the degree of feature x, d_s is the degree of sensible feature, v_s is the value of sensible feature, β is the relative effectiveness factor between sensible feature and other features. Note that β takes a value in the range of [0, 0.2]. The mathematical equation $\beta(d_s - d_x)$ is used to adjust quantity amounts between sensible feature and other features.

4.3.2 Behavioral inference

Behavioral inference is the most important unit of the proposed decision system. It is, thereby, designed more similar to humanistic reasoning to make appropriate decisions under various behavioral conditions. Obtaining the various behavioral styles, aggregating the behavioral styles together, producing a total behavior, and specifying a success set based on humanistic behavioral reasoning are performed by the behavioral inference. This unit is composed of four sub-units including behaviors collection, total behavior, behavioral aggregation, and inference as explained in the following subsections.

4.3.2.1. Behaviors collection After both of importance factors and belongingness factors of the futures are determined by manual way or one of the decomposition methods, any behavioral style is stored in a matrix called behavioral matrix. This matrix consists of n rows and three columns where n is the number of features in feature collection. Each row indicates the information related to one of the features as well as each column indicates three attributes 'degree', 'value' and 'truth'. Structure of the behavioral matrix is shown in Fig. 9 where the set of $\{d_1, d_2, d_3, \ldots, d_n\}$ represents attribute 'degree', the set of $\{v_1, v_2, v_3, \ldots, v_n\}$ represents attribute 'value', and the set of $\{t_1, t_2, t_3, \ldots, t_n\}$ represents attribute 'truth'. Attribute

	degree	value	truth
1	d_1	v_1	t_1
2	d_2	v_2	t_2
3	d_3	v_3	t_3
...	...	...	...
n	d_n	v_n	t_n

Fig. 9 Structure of the behavioral matrix

	degree	truth	growth
1	d_1	t_1	g_1
2	d_2	t_2	g_2
3	d_3	t_3	g_3
...	...	...	...
n	d_n	t_n	g_n

Fig. 10 Structure of the total matrix

'degree' is equal to the degree of any feature in behavioral set and attribute 'value' is equal to the value of any feature in behavioral set. Furthermore, attribute 'truth' is calculated based on attribute 'value' and a trust amount as follow

$$\forall x \in n,\ t(x) = v(x) * \gamma \tag{7}$$

where x indicates identifier number of each feature in feature collection F, n indicates the number of features, v(x) represents the value of feature x in behavioral set, and γ indicates a trust amount in the range of [0, 1]. Whereas various behavioral styles are obtained through different ways, their trust amounts are not the same. If they are obtained by the system itself, the trust amount is usually equal to 1; otherwise, the trust amount takes a value in the range of [0, 1]. The trust amount is used in the behavioral matrix to specify a special effectiveness weight to any obtained behavioral style. For example, the behavioral style received from a family member is more reliable than that of received from a friend. Behaviors collection is composed of several behavioral matrixes that are built by various behavioral styles. Note that all behavioral matrixes are aggregated together to produce a total behavior as described in the following sub-sections.

4.3.2.2. Total behavior Various behavioral matrixes determined by the system are aggregated together by one of the proposed behavioral aggregation methods, those of described in Sect. 4.3.2.3, to produce a total behavior based on the obtained knowledge base. The total behavior is stored in a matrix called total matrix which consists of n rows and three columns where n is the number of features. Each row represents the information about one of the features as well as each column indicates three attributes 'degree', 'truth' and 'growth'. Upon the behavioral matrix of a new behavioral set is determined by BDS, it is aggregated with current total matrix to update behavioral information of the total matrix based on the up-to-date knowledge base. Structure of the total matrix is shown in Fig. 10 where the set of $\{d_1, d_2, d_3, \ldots, d_n\}$ represents attribute 'degree', the set of $\{t_1, t_2, t_3, \ldots, t_n\}$ represents attribute 'truth', and the set of $\{g_1, g_2, g_3, \ldots, g_n\}$ represents attribute 'growth'. If the behavioral matrix determined by the system is associated to the first behavioral style, attribute 'degree' of the total matrix is equal to attribute 'degree' of the behavioral matrix as well as attributes 'truth' and 'growth' of the total matrix are equal to attribute 'truth' of the behavioral matrix; otherwise, all attributes of the total matrix are calculated by one of the proposed behavioral aggregation methods described in Sect. 4.3.2.3.

In some conditions, we need to store several total behaviors associated to various people and smart systems, separately, to select the best one to accomplish a desired behavioral style. In this case, their behavioral styles are obtained by the system, separately, to update their dedicated total matrixes to specify a separate success rate for each of them. For example, if there are three drivers to accomplish a driving job, their prior behavioral styles are entered in their knowledge bases to construct their dedicated total matrixes. Afterward, a success rate is specified for each of the drivers based on their total matrixes and the desired behavioral style. Finally, the driver who has the highest success rate is selected to do the job. This procedure can be used in most of the engineering applications and control systems to select appropriate people and smart systems, correctly.

4.3.2.3. Behavioral aggregation Upon a new behavioral style is obtained by BDS, it is aggregated with the total matrix to update the total behavior. This process is performed by one of the proposed behavioral aggregation methods described in this sub-section. Some humans have a pessimistic thinking, someone have an optimistic thinking, and some others have a fair thinking in face of various environmental conditions. These thinking types are used in the proposed aggregation methods in a way that the proposed decision system can make appropriate decisions based on the various humanistic reasoning. The behavioral aggregation process between a behavioral style and the total behavior is denoted by the symbol $\uplus$. The algebraic form of the behavioral aggregation is represented as follow

$$T = B \uplus T \tag{8}$$

where B indicates behavioral set of the behavioral style and T indicates the total matrix. The proposed behavioral aggregation methods are categorized into three groups: pessimistic behavioral aggregation, optimistic behavioral aggregation, and fair behavioral aggregation.

4.3.2.3.1. Pessimistic behavioral aggregation Some humans tend to see the worst aspect of things or believe that the worst things will be happened. An aggregation method called pessimistic behavioral aggregation is proposed in this subsection that uses the worst reasoning type. It can be properly used to develop and design more precise systems. The algebraic form of the pessimistic aggregation is represented as follow

$$\forall x \in n, \begin{cases} T_x^1 = \frac{B_x^1 + T_x^1}{2} \\ T_x^2 = min\left(B_x^3, T_x^2\right) \\ T_x^3 = min\left(B_x^3, T_x^2, T_x^3\right) \end{cases} \tag{9}$$

where x indicates identifier number of each feature in feature collection F, n indicates the number of features, T_x^1 represents the degree of feature x in the total matrix, T_x^2 represents the truth of feature x in the total matrix, T_x^3 represents the growth of feature x in the total matrix, B_x^1 represents the degree of feature x in the behavioral set, and B_x^3 represents the truth of feature x in the behavioral set. Because the worst reasoning type is considered by this aggregation method, the second attribute of the total matrix is set by the minimum value of attributes ‘truth’ related to the behavioral set and the total matrix. Furthermore, the third attribute of the total matrix is calculated based on the minimum value of three quantity amounts including the truth of the features in the behavioral set, the truth of the features in the total matrix, and the growth of the features in the total matrix.

4.3.2.3.2. Optimistic behavioral aggregation Some humans have the hopeful and confident aspect about the future. They tend to see the best aspect of things or believe that the best things will be happened. This type of humanistic thinking is used in the proposed decision system to aggregate various behavioral sets together by an aggregation method called optimistic behavioral aggregation. This method can be used to develop and design those of the systems which are not very accurate. The algebraic form of the optimistic aggregation is represented as follow

$$\forall x \in n, \begin{cases} T_x^1 = \frac{B_x^1 + T_x^1}{2} \\ T_x^2 = max\left(B_x^3, T_x^2\right) \\ T_x^3 = max\left(B_x^3, T_x^2, T_x^3\right) \end{cases} \tag{10}$$

where x is identifier number of each feature in feature collection F, n is the number of features, T_x^1 is the degree of feature x in the total matrix, T_x^2 is the truth of feature x in the total matrix, T_x^3 is the growth of feature x in the total matrix, B_x^1 is the degree of feature x in the behavioral set, and B_x^3 is the truth of feature x in the behavioral set. Whereas the best thinking type is considered by the optimistic aggregation method, the second attribute of the total matrix is set by the maximum value of attributes ‘truth’ related to the behavioral set and the total matrix. Moreover, the third attribute of the total matrix is determined by the maximum value of three quantity amounts including the truth of the features in the behavioral set, the truth of the features in the total matrix, and the growth of the features in the total matrix.

4.3.2.3.3. Fair behavioral aggregation In contrast to the worst aspect and the best aspect of things, some humans have the fair aspect about the future. The fair reasoning type is utilized by the proposed decision system to offer an aggregation method called fair behavioral aggregation. This method can be used to develop and design both of the more precise systems and the less accurate systems. The algebraic form of the fair aggregation is represented as follow

$$\forall x \in n, \begin{cases} T_x^1 = \frac{B_x^1 + T_x^1}{2} \\ T_x^2 = \frac{B_x^3 + T_x^2}{2} \\ T_x^3 = \frac{\left(\frac{min\left(B_x^3, T_x^2, T_x^3\right) + max\left(B_x^3, T_x^2, T_x^3\right)}{2}\right) + T_x^3}{2} \end{cases} \tag{11}$$

where x represents identifier number of each feature in feature collection F, n represents the number of features, T_x^1 indicates the degree of feature x in the total matrix, T_x^2 indicates the truth of feature x in the total matrix, T_x^3 indicates the growth of feature x in the total matrix, B_x^1 indicates the degree of feature x in the behavioral set, and B_x^3 indicates the truth of feature x in the behavioral set. As represented in the above mathematical equations, the mean quantity amounts of the attributes are considered to update the total matrix in a fair procedure.

4.3.2.4. Inference As represented before, the proposed BBDM method is designed to make appropriate decisions based on humanistic behavioral reasoning in face of different environmental conditions. When a desired behavioral style is requested from the system, a success set is produced by the inference unit. This is done based on the desired style and the total behavior. In fact, the success set forecasts an expected behavior from the system according to the obtained knowledge base. Any desired behavioral style is denoted by a set called desired set. This set includes attribute 'degree' to specify the required degrees of the features. Attribute 'degree' determines effectiveness weight of any feature in the desired set that is called importance factor. The algebraic form of the desired set is represented as follow

$$D = \left\{ \frac{\eta_D(x)}{x} \mid x \in F \right\} \tag{12}$$

where x is identifier number of each feature in feature collection F and $\eta_D(x)$ is the importance factor of feature x in set D. Note that degrees of the features in set D are determined by manual way. The success set consists of attributes 'degree' and 'value' to specify the degrees and values of the features. Attribute 'degree' determines effectiveness weight of any feature in the success set that is called importance factor. Attribute 'value' determines quantity amount of any feature in the success set that is called belongingness factor. The algebraic form of the success set is represented as follow

$$S = \left\{ \frac{(\eta_S(x), \partial_S(x))}{x} \mid x \in F \right\} \tag{13}$$

where x is identifier number of each feature in feature collection F, $\eta_S(x)$ is the importance factor of feature x in set S, and $\partial_S(x)$ is belongingness factor of the feature x in set S. Importance factors of the features in success set are determined based on importance factors of the features in desired set as follow

$$\forall x \in n, S_x^1 = D_x \tag{14}$$

where x is identifier number of each feature in feature collection F, n is the number of features, and D_x is the importance factor of feature x in desired set D. Furthermore, belongingness factors of the features in success set are calculated by a behavioral inference process between the desired set and the total matrix. The inference process is denoted by the symbol $\circledcirc$. The algebraic form of the inference unit to specify belongingness factors of the success set is represented as follow

$$S = D \circledcirc T \tag{15}$$

where D represents the desired set and T represents the total matrix. The inference process is performed by one of the proposed inference methods described in the following subsections. These methods are categorized into three groups: straight inference, relative inference, and impartial inference.

4.3.2.4.1. Straight inference Straight inference is one of the humanistic behavioral reasoning methods to determine belongingness factors of the features in success set. It is also called fair inference because an equitable thinking is considered among all features. In this inference type, belongingness factors of the features in the success set are calculated via the comparison of peer to peer features in the desired set and the total matrix. If a driving behavior is considered here with several features (e.g., safety), the following statements are some instances of the straight inference:

- If total matrix involves low safety, the driver will accomplish a driving style with low safety.
- If total matrix involves low safety, the driver will not accomplish a driving style with high safety.
- If total matrix involves high safety, the driver will accomplish a driving style with low safety.
- If total matrix involves high safety, the driver will accomplish a driving style with high safety.

The above statements represent that if a human pays attention to a special feature, he will always pay attention to that feature. The algebraic form of the straight inference is represented as follow

$$\forall x \in n, S_x^2 = \begin{cases} \varphi \left| T_x^2 - \sigma \right| + \omega T_x^3, & D_x \le T_x^1 \\ \varphi (T_x^2 + \sigma) + \omega\, T_x^3, & D_x > T_x^1 \end{cases} \tag{16}$$

where x represents identifier number of each feature in feature collection F, n represents the number of features, T_x^1 represents the degree of feature x in the total matrix, T_x^2 represents the truth of feature x in the total matrix, T_x^3 represents the growth of feature x in the total matrix, D_x represents the importance factor of feature x in desired set D, σ represents the difference between quantity amounts of the desired set and the total matrix that takes a value in the range of (0, 0.2], φ represents the impact rate of attribute 'truth' related to the total matrix that takes a value in the range of [0, 1], and ω represents the impact rate of attribute 'growth' related to the total matrix that takes the value of $1 - \varphi$.

4.3.2.4.2. Relative inference Relative inference is another humanistic inference to calculate belongingness factors of the features in success set. In this type, belongingness factors are determined based on a relation between features of the desired set and the total matrix. It is also called inverse inference because belongingness factors of the features are specified via an inverse relation between various behavioral styles. If a driving behavior is assumed here with several features (e.g., safety), the following statements represent some instances of the relative inference:

- If total matrix involves low safety, the driver will not accomplish a driving style with high safety.
- If total matrix does not involve low safety, the driver will accomplish a driving style with high safety.

- If total matrix involves high safety, the driver will not accomplish a driving style with low safety.
- If total matrix does not involve high safety, the driver will accomplish a driving style with low safety.

The above statements describe the humanistic inverse reasoning to predict some of the unforeseen conditions. The algebraic form of the relative inference is represented as follow

$$\forall x \in n,\ S_x^2 = \begin{cases} \varphi \frac{D_x * T_x^2}{T_x^1} + \omega T_x^3, & D_x \leq T_x^1 \\ \varphi \left(\frac{D_x * T_x^2}{T_x^1} + T_x^2 \right) + \omega T_x^3, & \left(D_x > T_x^1\right) \ \&\ \left(T_x^2 \leq 0.5\right) \\ 1, & \left(D_x > T_x^1\right) \ \&\ \left(T_x^2 > 0.5\right) \end{cases} \tag{17}$$

where x indicates identifier number of each feature in feature collection F, n indicates the number of features, T_x^1 indicates the degree of feature x in the total matrix, T_x^2 indicates the truth of feature x in the total matrix, T_x^3 indicates the growth of feature x in the total matrix, D_x indicates the importance factor of feature x in desired set D, φ indicates the impact rate of attribute 'truth' related to the total matrix that takes a value in the range of [0, 1], and ω indicates the impact rate of attribute 'growth' related to the total matrix that takes the value of 1 − φ.

4.3.2.4.3. Impartial inference Both of the straight and relative inferences determine belongingness factors of the features in success set based on a sensitive thinking without considering any balanced reasoning. They cause belongingness factors not to be, correctly, specified in face of some environmental conditions. A weighted reasoning method called impartial inference is proposed here to calculate belongingness factors of the features in success set. The algebraic form of the impartial inference is represented as follow

$$\forall x \in n,\ S_x^2 = \begin{cases} \varphi \frac{D_x * T_x^2}{\sum_{y=1}^{n} \left(T_y^1 * T_y^2\right)} + \omega \frac{D_x * T_x^3}{\sum_{y=1}^{n} \left(T_y^1 * T_y^3\right)}, & D_x \leq T_x^1 \\ min \left(\left(\varphi \frac{D_x * T_x^2}{\sum_{y=1}^{n} \left(T_y^1 * T_y^2\right)} + \omega \frac{D_x * T_x^3}{\sum_{y=1}^{n} \left(T_y^1 * T_y^3\right)} \right), 1 \right), & D_x > T_x^1 \end{cases} \tag{18}$$

where x is identifier number of each feature in feature collection F, y is identifier number of each feature in feature collection F, n is the number of features, T_x^1 is the degree of feature x in the total matrix, T_x^2 is the truth of feature x in the total matrix, T_x^3 is the growth of feature x in the total matrix, T_y^1 is the degree of feature y in the total matrix, T_y^2 is the truth of feature y in the total matrix, T_y^3 is the growth of feature y in the total matrix, D_x is the importance factor of feature x in desired set D, φ is the impact rate of attribute 'truth' related to the total matrix that takes a value in the range of [0, 1], and ω is the impact rate of attribute 'growth' related to the total matrix that takes the value of 1 − φ.

4.3.3 Composition

Since real world only deals with real numbers, success set cannot be used in actual systems and applications. Therefore, this set must be converted to a crisp value in the proposed decision system. The composition unit is defined to convert any success set to a success rate in the range of [0, 1]. The proposed composition methods are categorized into three groups: min-degree composition, max-degree composition, and weighted composition.

4.3.3.1. Min-degree composition In a composition type, success rate is equal to belongingness factor of the feature which has the lowest importance factor in the success set. This method, called min-degree composition, can be used to specify the success rate in those of the systems which are not very accurate. Because it is possible that more than one feature have the lowest importance factor in the success set, average of their belongingness factors is calculated by the method. The algebraic form of the min-degree composition is represented as follow

$$R = \frac{\sum_{x=1}^{n} S_x^2}{n} \tag{19}$$

where x indicates identifier number of each feature which has the lowest importance factor in the success set, n indicates the number of the features which have the lowest importance factor, and S_x^2 represents belongingness factor of feature x in the success set.

4.3.3.2. Max-degree composition In some cases (e.g., in more precise systems), belongingness factor of the feature which has the highest importance factor in the success set is selected as the success rate. This composition type, called max-degree composition, can be used when an accurate reasoning is requested from the system. Since it is possible that more than one feature have the highest importance factor in the success set, the mean value of their belongingness factors is considered by the method. The algebraic form of the max-degree composition is represented as follow

$$R = \frac{\sum_{y=1}^{n} S_y^2}{n} \tag{20}$$

where y is identifier number of each feature which has the highest importance factor in the success set, n is the number of the features which have the highest importance factor, and S_y^2 is belongingness factor of feature y in the success set.

4.3.3.3. Weighted composition Weighted composition considers a fair behavior between all features of the success set that can be used in most of control applications and systems. In this composition type, importance factors and belongingness factors of all features are participated to calculate the success rate. That is, it works based on quantity amounts of all features instead of considering quantity amounts of some special features. The algebraic form of the weighted composition is represented as follow

$$R = \sum_{x=1}^{n} \left(S_x^1 * S_x^2 \right) \tag{21}$$

where x represents identifier number of each feature in feature collection F, n represents the number of all features, S_x^1 represents importance factor of feature x in the success set, and S_x^2 represents belongingness factor of feature x in the success set.

The proposed decision system has a learning engine to obtain, any time, the results of desired behavioral styles. After the success rate of any desired behavioral style is calculated by the system and that behavior is accomplished by a human or a smart system, the results of the completed behavior can be obtained by the system. This is done to update the total behavior for increasing the reliability of the proposed system over time. The learning engine can solve some possible problems of the system via gathering new behavioral information.

5 Explaining the BBDM method with a practical example

This section represents a practical example related to the driving behavior to explain behavior-based decision making (BBDM) method. Various behavioral styles are obtained by behavioral decision system (BDS) to construct the total matrix. When any desired behavioral style is requested from the system, BDS produces a success rate based on the obtained knowledge base to forecast the success probability of the system. Suppose for this example, the style collection is defined as SC ={ 'safe', 'accurate', 'medium', 'on time', 'safe & fast', 'accurate & fast'} and the feature collection is defined as F = { 'safety', 'accuracy', 'speed', 'timing'} . Four initial behavioral styles are obtained by the system as represented in Table 1. Because B_1 and B_2 are collected by the system itself, their trust amounts are equal to 1. Whereas B_2 and B_3 are reported from other humans or smart system, their trust amounts are considered as 0.9 and 0.7, respectively.

The importance factors of the features in the first obtained behavioral style, denoted by 'safe', are determined by decremental importance function and the belongingness factors of the features are determined by manual way. The behavioral set of the first obtained style is represented as follow

Table 1 The initial behavioral styles obtained by BDS

#	Behavioral style	Trust amount (γ)
1	Safe	1
2	Accurate and fast	1
3	Medium	0.9
4	On time	0.7

$$B_1 = \left\{ \frac{(0.4,\ 1)}{\text{safety}}, \frac{(0.3,\ 0.6)}{\text{accuracy}}, \frac{(0.2,\ 0.7)}{\text{speed}}, \frac{(0.1,\ 0.2)}{\text{timing}} \right\}$$

The importance factors of the features in the second obtained behavioral style, denoted by 'accurate & fast', are specified by decremental importance function and the belongingness factors of the features are specified by manual way. The behavioral set of the second obtained style is represented as follow

$$B_2 = \left\{ \frac{(0.4,\ 0.7)}{\text{safety}}, \frac{(0.3,\ 1)}{\text{accuracy}}, \frac{(0.2,\ 1)}{\text{speed}}, \frac{(0.1,\ 0.3)}{\text{timing}} \right\}$$

The importance factors of the features in the third obtained behavioral style, denoted by 'medium', are determined by identical importance function where the number of decimal places is equal to 2. Furthermore, the belongingness factors of the features in this behavioral style are determined by inverse belongingness function where the feature 'safety' is considered as the sensible feature, the value of the sensible feature is equal to 0.9, and the parameter α is equal to 0.1. The behavioral set of the third obtained style is represented as follow

$$B_3 = \left\{ \frac{(0.25,\ 0.9)}{\text{safety}}, \frac{(0.25,\ 0.9)}{\text{accuracy}}, \frac{(0.25,\ 0.9)}{\text{speed}}, \frac{(0.25,\ 0.9)}{\text{timing}} \right\}$$

The importance factors of the features in the fourth obtained behavioral style, denoted by 'on time', are specified by incremental importance function. Furthermore, the belongingness factors of the features in this behavioral style are specified by relativism belongingness function where the feature 'speed' is considered as the sensible feature, the value of the sensible feature is equal to 0.4, and the parameter β is equal to 0.05. The behavioral set of the fourth obtained style is represented as follow

$$B_4 = \left\{ \frac{(0.1,\ 0.39)}{\text{safety}}, \frac{(0.2,\ 0.395)}{\text{accuracy}}, \frac{(0.3,\ 0.4)}{\text{speed}}, \frac{(0.4,\ 0.405)}{\text{timing}} \right\}$$

After behavioral sets of the initial behavioral styles are determined by the decomposition unit, behavioral matrixes of the styles are constructed in the behavioral inference unit. The behavioral matrixes of the initial behavioral styles are represented as follows

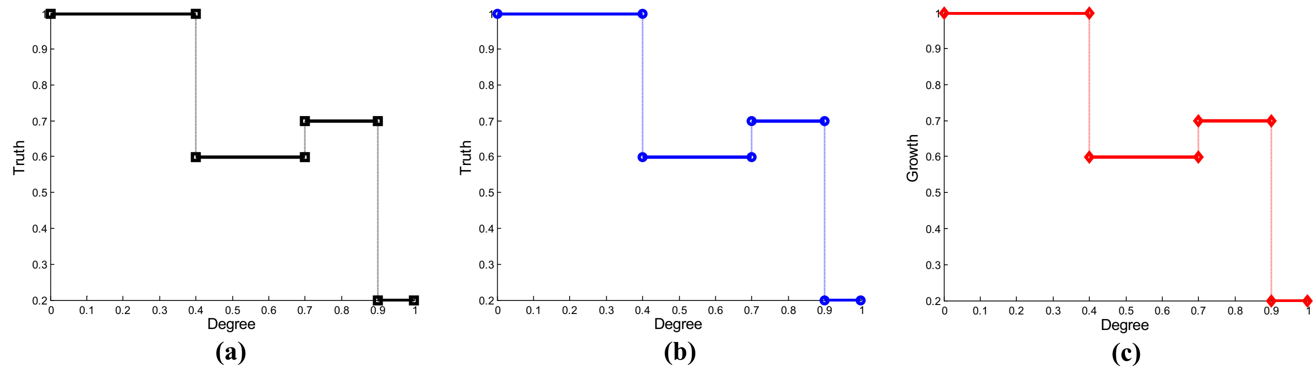


Fig. 11 Effectiveness weight of the behavioral matrix BM_1 on the total behavior. **a** Attribute 'truth' of the behavioral matrix BM_1 versus attribute 'degree', **b** attribute 'truth' of the total matrix versus attribute 'degree', **c** attribute 'growth' of the total matrix versus attribute 'degree'

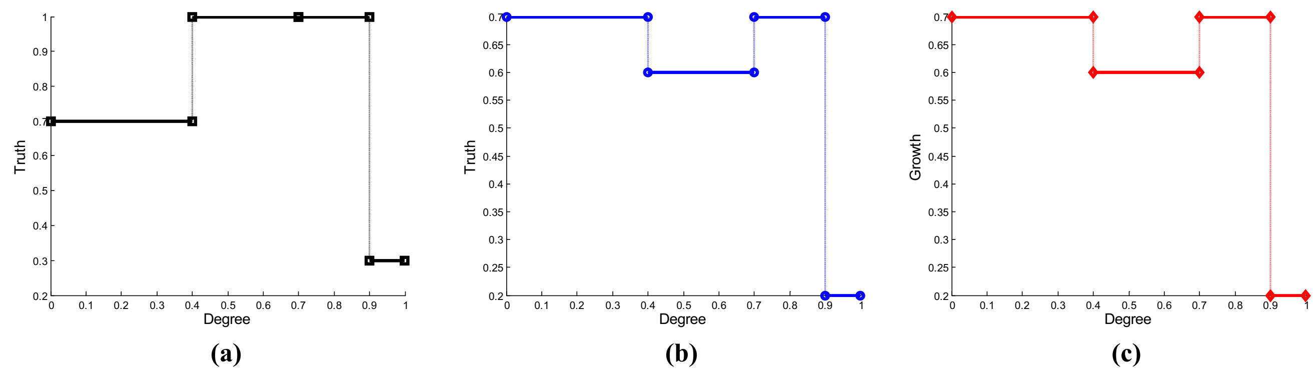


Fig. 12 Effectiveness weight of the behavioral matrix BM_2 on the total behavior. **a** Attribute 'truth' of the behavioral matrix BM_2 versus attribute 'degree', **b** attribute 'truth' of the total matrix versus attribute 'degree', **c** Attribute 'growth' of the total matrix versus attribute 'degree'

$$BM_1 = \begin{bmatrix} 0.4 \; 1.0 \; 1.0 \\ 0.3 \; 0.6 \; 0.6 \\ 0.2 \; 0.7 \; 0.7 \\ 0.1 \; 0.2 \; 0.2 \end{bmatrix} \quad BM_2 = \begin{bmatrix} 0.4 \; 0.7 \; 0.7 \\ 0.3 \; 1.0 \; 1.0 \\ 0.2 \; 1.0 \; 1.0 \\ 0.1 \; 0.3 \; 0.3 \end{bmatrix}$$

$$BM_3 = \begin{bmatrix} 0.25 \; 0.9 \; 0.81 \\ 0.25 \; 0.9 \; 0.81 \\ 0.25 \; 0.9 \; 0.81 \\ 0.25 \; 0.9 \; 0.81 \end{bmatrix} \quad BM_4 = \begin{bmatrix} 0.1 \; 0.390 \; 0.273 \\ 0.2 \; 0.395 \; 0.277 \\ 0.3 \; 0.400 \; 0.280 \\ 0.4 \; 0.405 \; 0.284 \end{bmatrix}$$

After behavioral matrixes of the behavioral sets are determined separately, the total matrix is produced by behavioral aggregation based on the behavioral matrixes in order to update the total behavior. Before any aggregation process in the system, there is no information in the total matrix as follow

$$T = \begin{bmatrix} 0 \; 0 \; 0 \\ 0 \; 0 \; 0 \\ 0 \; 0 \; 0 \\ 0 \; 0 \; 0 \end{bmatrix}$$

Because BM_1 is the behavioral matrix of the first behavioral style, it is copied into the total matrix without any behavioral aggregation process as follow

$$T = \begin{bmatrix} 0.4 \; 1.0 \; 1.0 \\ 0.3 \; 0.6 \; 0.6 \\ 0.2 \; 0.7 \; 0.7 \\ 0.1 \; 0.2 \; 0.2 \end{bmatrix} \dot{\cup} \begin{bmatrix} 0 \; 0 \; 0 \\ 0 \; 0 \; 0 \\ 0 \; 0 \; 0 \\ 0 \; 0 \; 0 \end{bmatrix}$$
$$= \begin{bmatrix} 0.4 \; 1.0 \; 1.0 \\ 0.3 \; 0.6 \; 0.6 \\ 0.2 \; 0.7 \; 0.7 \\ 0.1 \; 0.2 \; 0.2 \end{bmatrix}$$

Effectiveness weight of the behavioral matrix BM_1 on the total behavior is shown in Fig. 11. Because BM_1 is the first behavioral matrix specified by the system, the attributes 'truth' and 'growth' of the total matrix are equal to the attribute 'truth' of the behavioral matrix BM_1.

It is assumed that the behavioral matrix BM_2 is aggregated with the total matrix by pessimistic behavioral aggregation. The total matrix is resulted after the aggregation process with BM_2 as follow

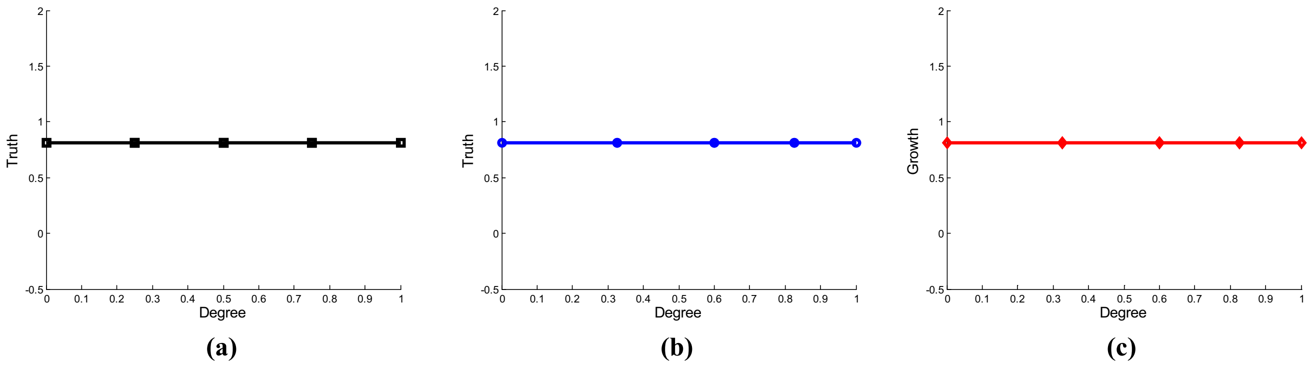


Fig. 13 Effectiveness weight of the behavioral matrix BM_3 on the total behavior. **a** Attribute 'truth' of the behavioral matrix BM_3 versus attribute 'degree', **b** attribute 'truth' of the total matrix versus attribute 'degree', **c** attribute 'growth' of the total matrix versus attribute 'degree'

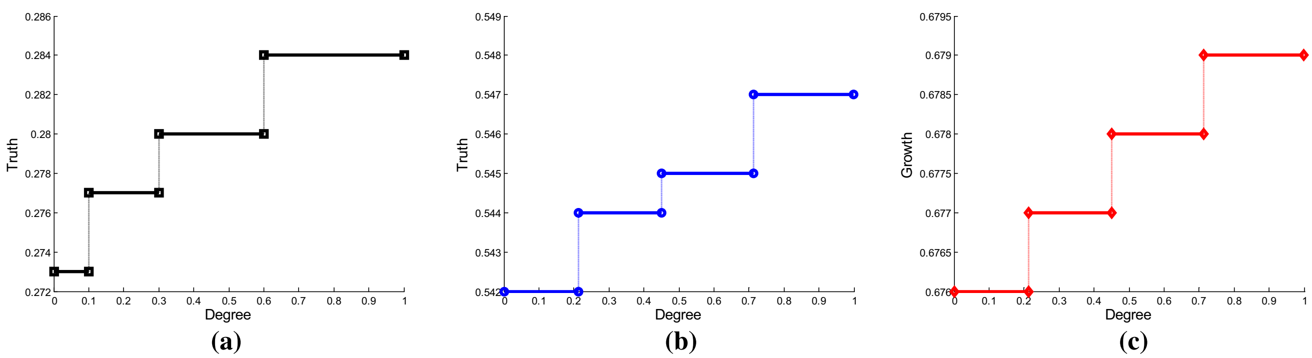


Fig. 14 Effectiveness weight of the behavioral matrix BM_4 on the total behavior. **a** Attribute 'truth' of the behavioral matrix BM_4 versus attribute 'degree', **b** attribute 'truth' of the total matrix versus attribute 'degree', **c** attribute 'growth' of the total matrix versus attribute 'degree'

$$T = \begin{bmatrix} 0.4 & 0.7 & 0.7 \\ 0.3 & 1.0 & 1.0 \\ 0.2 & 1.0 & 1.0 \\ 0.1 & 0.3 & 0.3 \end{bmatrix} \dot{\cup} \begin{bmatrix} 0.4 & 1.0 & 1.0 \\ 0.3 & 0.6 & 0.6 \\ 0.2 & 0.7 & 0.7 \\ 0.1 & 0.2 & 0.2 \end{bmatrix} = \begin{bmatrix} 0.4 & 0.7 & 0.7 \\ 0.3 & 0.6 & 0.6 \\ 0.2 & 0.7 & 0.7 \\ 0.1 & 0.2 & 0.2 \end{bmatrix}$$

Effectiveness weight of the behavioral matrix BM_2 on the total behavior is shown in Fig. 12. As illustrated by the simulation results, quantity amounts of the features in the total behavior are grown considerably because quantity amounts of the attribute 'truth' in the behavioral matrix BM_2 are high.

It is considered that the behavioral matrix BM_3 is aggregated with the total matrix by optimistic behavioral aggregation. The total matrix is updated after the aggregation process with BM_3 as follow

$$T = \begin{bmatrix} 0.25 & 0.9 & 0.81 \\ 0.25 & 0.9 & 0.81 \\ 0.25 & 0.9 & 0.81 \\ 0.25 & 0.9 & 0.81 \end{bmatrix} \dot{\cup} \begin{bmatrix} 0.4 & 0.7 & 0.7 \\ 0.3 & 0.6 & 0.6 \\ 0.2 & 0.7 & 0.7 \\ 0.1 & 0.2 & 0.2 \end{bmatrix}$$
$$= \begin{bmatrix} 0.325 & 0.81 & 0.81 \\ 0.275 & 0.81 & 0.81 \\ 0.225 & 0.81 & 0.81 \\ 0.175 & 0.81 & 0.81 \end{bmatrix}$$

Effectiveness weight of the behavioral matrix BM_3 on the total behavior is shown in Fig. 13. As depicted by the simulation results, quantity amounts of the features in the total behavior are become uniform after the behavioral aggregation process between the current total matrix and the behavioral matrix BM_3.

It is supposed that the behavioral matrix BM_4 is aggregated with the total matrix by fair behavioral aggregation. The total matrix is resulted after the aggregation process with BM_4 as follow

$$T = \begin{bmatrix} 0.1 & 0.390 & 0.273 \\ 0.2 & 0.395 & 0.277 \\ 0.3 & 0.400 & 0.280 \\ 0.4 & 0.405 & 0.284 \end{bmatrix} \dot{\cup} \begin{bmatrix} 0.325 & 0.81 & 0.81 \\ 0.275 & 0.81 & 0.81 \\ 0.225 & 0.81 & 0.81 \\ 0.175 & 0.81 & 0.81 \end{bmatrix}$$
$$= \begin{bmatrix} 0.2125 & 0.542 & 0.676 \\ 0.2375 & 0.544 & 0.677 \\ 0.2625 & 0.545 & 0.678 \\ 0.2875 & 0.547 & 0.679 \end{bmatrix}$$

Effectiveness weight of the behavioral matrix BM_4 on the total behavior is shown in Fig. 14. The simulation results illustrate that the behavioral matrix BM_4 causes quantity amounts of the attributes 'truth' and 'growth' in the total behavior to be decreased considerably. The reason is that quantity amounts of the attribute 'truth' in the behavioral matrix BM_4 are low and the fair behavioral aggregation is used to aggregate BM_4 with the total matrix.

The latest total matrix is the total behavior of the system based on the obtained knowledge base. When any desired behavioral style is requested from the system, a success set is produced for the desired style based on the total matrix with the aid of the inference process. It is supposed that three desired behavioral styles 'safe & fast', 'accurate', and 'accurate & fast' are requested from the system. These styles are represented by the desired sets D_1, D_2, and D_3, respectively. The importance factors of the features in the desired sets are specified by manual way as follows

$$D_1 = \left\{ \frac{0.9}{\text{safety}}, \frac{0.1}{\text{accuracy}}, \frac{0.7}{\text{speed}}, \frac{0.15}{\text{timing}} \right\}$$
$$D_2 = \left\{ \frac{0.1}{\text{safety}}, \frac{0.9}{\text{accuracy}}, \frac{0.4}{\text{speed}}, \frac{0.15}{\text{timing}} \right\}$$
$$D_3 = \left\{ \frac{0.1}{\text{safety}}, \frac{0.5}{\text{accuracy}}, \frac{0.7}{\text{speed}}, \frac{0.2}{\text{timing}} \right\}$$

The importance factors of the features in the success sets are equal to the importance factors of the features in the desired sets as follows

$$S_1 = \left\{ \frac{(0.9,\ 0.0)}{\text{safety}}, \frac{(0.1,\ 0.0)}{\text{accuracy}}, \frac{(0.7,\ 0.0)}{\text{speed}}, \frac{(0.15,\ 0.0)}{\text{timing}} \right\}$$
$$S_2 = \left\{ \frac{(0.1,\ 0.0)}{\text{safety}}, \frac{(0.9,\ 0.0)}{\text{accuracy}}, \frac{(0.4,\ 0.0)}{\text{speed}}, \frac{(0.15,\ 0.0)}{\text{timing}} \right\}$$
$$S_3 = \left\{ \frac{(0.1,\ 0.0)}{\text{safety}}, \frac{(0.5,\ 0.0)}{\text{accuracy}}, \frac{(0.7,\ 0.0)}{\text{speed}}, \frac{(0.2,\ 0.0)}{\text{timing}} \right\}$$

The belongingness factors of the features in the success sets are specified by the behavioral inference process between the desired sets and the total matrix. It is assumed that the belongingness factors of S_1 are determined by the straight inference method where the parameter σ is equal to 0.2, the parameter φ is equal to 0.4, and the parameter ω is equal to 0.6. Therefore, the success set S_1 is represented after the inference process as follow

$$S_1 = \left\{ \frac{(0.9,\ 0.702)}{\text{safety}}, \frac{(0.1,\ 0.544)}{\text{accuracy}}, \frac{(0.7,\ 0.705)}{\text{speed}}, \frac{(0.15,\ 0.546)}{\text{timing}} \right\}$$

It is supposed that the belongingness factors of S_2 are specified by the relative inference method where the parameter φ is equal to 0.4 and the parameter ω is equal to 0.6. Therefore, the success set S_2 is resulted after the inference process as follow

$$S_2 = \left\{ \frac{(0.1,\ 0.508)}{\text{safety}}, \frac{(0.9,\ 1)}{\text{accuracy}}, \frac{(0.4,\ 1)}{\text{speed}}, \frac{(0.15,\ 0.522)}{\text{timing}} \right\}$$

It is considered that the belongingness factors of S_3 are calculated by the impartial inference method where the parameter φ is equal to 0.4 and the parameter ω is equal to 0.6. Therefore, the success set S_3 is resulted after the inference process as follow

$$S_3 = \left\{ \frac{(0.1,\ 0.1)}{\text{safety}}, \frac{(0.5,\ 0.5)}{\text{accuracy}}, \frac{(0.7,\ 0.7)}{\text{speed}}, \frac{(0.2,\ 0.2)}{\text{timing}} \right\}$$

After the success sets of the desired behavioral styles are calculated by the behavioral inference process, they are converted to three success rates by one of the proposed composition methods. It is assumed that min-degree composition is used to convert the success set S_1 to the success rate R_1, max-degree composition is applied to convert the success set S_2 to the success rate R_2, and weighted composition is used to convert the success set S_3 to the success rate R_3. Therefore, the success rates of the desired behavioral styles are resulted as follows

$$R_1 = 0.544$$
$$R_2 = 1.000$$
$$R_3 = 0.790$$

The above success rates represent that the success probability of the system to accomplish the first desired behavioral style is medium, the success probability of the system to accomplish the second desired behavioral style is very high, and the success probability of the system to accomplish the third desired behavioral style is high.

6 Discussion and analysis results

BBDM uses humanistic behavioral reasoning to make appropriate decisions under various conditions. The main advantages of the BBDM method compared to some of the existing intelligent methods are represented in Table 2. Discussion and analysis process is resulted in terms of decision making type, humanistic data reasoning, humanistic behavioral reasoning, public knowledge base, private knowledge base, reasoning type, decision making for systems, and decision making for humans. As described before, the existing intelligent methods such as ANNs, machine learning, MCDM, fuzzy logic, evolutionary computing, learning theory, and deep learning in neural networks are presented to apply some of the intelligent mechanisms for appropriate decision making. Although these methods are potential methods involving the significant advantages, they have some of the major behavioral constraints (e.g., the lack of humanistic behaviors) so that they cannot be completely used in the most of humanistic environments. In contrast, the proposed BBDM

Table 2 Analysis results of the BBDM method compared to some intelligent methods

Method name	Analysis terms							
	Decision making type	Humanistic data reasoning	Humanistic behavioral reasoning	Public knowledge base	Private knowledge base	Reasoning type	Decision making for systems	Decision making for humans
ANNs	Static decision making	Based on discrete states	Not capable	Yes	No	Static, but some of the ANNs are not static	Yes, but limited to some discrete states	No
Machine learning	Static decision making	Based on discrete states	Not capable	Yes	No	Static	Yes, but limited to some discrete states	No
MCDM	Based on a pre-defined database	Based on discrete conditions	Not capable	Yes	No	Usually static	Yes	No
Fuzzy logic	Based on a pre-defined database	Based on continuous states	Not capable	Yes	No	Usually static	Yes	No
Evolutionary computing	No	No	Not capable	No	No	The lack of any reasoning process	Based on random data generation	Based on random data generation
Learning theory	Static decision making	Based on static and limited reasoning	Capable based on static conditions without any reasoning process	Yes	No	Static	Not capable	Based on static and limited decisions
Deep learning in neural networks	Static decision making	Based on discrete states	Not capable	Yes	No	Static, but some of the NNs are not static	Yes, but limited to some discrete states	No
Bayesian network	Static decision making	Based on static and limited reasoning	Not capable	Yes	No	Static	Yes	No
ANFIS	Based on a pre-defined database	Based on continuous states	Not capable	Yes	No	Usually static	Yes	No
BBDM	Based on a pre-defined behavioral database	Based on potential behavioral states	Capable	Yes	Yes	Dynamic	Capable based on potential characteristics	Capable based on powerful properties

Table 3 Simulation parameters

Parameter	Value
Simulation time (seconds)	3600
The number of service providers	5
Buffer space of service providers (requests)	5
Minimum number of cores	1
Maximum number of cores	6
Service time (seconds)	100
Number of requests (requests)	10000
The number of customers	10
Φ	2
μ	0.3
α	0.05
β	0.1
σ	0.2
φ	0.8
ω	0.2

method uses behavior-based decision making to solve some of the existing problems in control and system engineering.

7 Simulation results

The proposed BBDM method can be used in various applications such as human-computer interaction, control, and engineering. In this section, it is compared to MCDM and fuzzy logic methods in the application for resource discovery in Cloud computing [31–33]. The simulations are carried out in a way that multiple Cloud service providers accomplish various requests of the customers in a Cloud-based system. The MCDM, fuzzy logic, and BBDM methods select an appropriate service provider to serve the customers. The simulation process is performed in 3600 seconds. The number of Cloud service providers is 5, the maximum buffer space of the providers is 5, and the number of CPU cores is varied from 1 to 6. Furthermore, the maximum service time of the requests is 100 seconds, the maximum number of requests is 10000, and the number of customers is 10. The requests are generated by Exponential distribution [34] where the parameter μ is equal to 0.3. Table 3 represents the systems parameters in details.

7.1 Simulation setup

In the BBDM method, the resource discovery behavior is considered in a way that style collection is defined by the list of SC = { 'slow', 'fast', 'light', 'heavy', 'poor', 'strong' } and feature collection is defined by the list of F = { 'number of cores', 'occupied buffer', 'service rate' } . The feature 'number of cores' is assumed as the sensible feature of the styles 'slow' and 'fast'; the feature 'occupied buffer' is assumed as the sensible feature of the styles 'light' and 'heavy'; finally, the feature 'service rate' is assumed as the sensible feature of the styles 'poor' and 'strong'. For all the service providers and all the behavioral styles, importance factors of the styles are specified by identical importance function, belongingness factors of the styles are determined by relativism belongingness function, the styles are aggregated together by fair behavioral aggregation, the inference process is performed by impartial inference, and composition process is done by weighted composition. Table 4 represents details of the initial knowledge bases and the total matrixes calculated for each Cloud service provider. Note that the Cloud service provider with the highest success rate is selected as an appropriate provider to accomplish the generated requests.

In the simulations, 'number of cores', 'occupied buffer', and 'service rate' are three input parameters of the MCDM and fuzzy logic methods as well as 'success probability' is the output parameter of these methods. The parameter 'number of cores' includes three linguistic terms (denoted by 'few', 'mediocre', and 'many'), the parameter 'occupied buffer' includes three linguistic terms (denoted by 'empty', 'partial', and 'full'), the parameter 'service rate' includes three linguistic terms (denoted by 'low', 'medium', and 'high'), and the parameter 'success probability' includes three linguistic terms (denoted by 'unsuccessful', 'moderate', and 'successful'). The MCDM and fuzzy logic methods use some pre-defined rules for decision making in the simulation process as presented in Table 5.

In the MCDM method, the linguistic term of the parameter 'number of cores' is determined as follow

$$N(x) = \begin{cases} few, & 1 \le x < 2 \\ mediocre, & 2 \le x < 4 \\ many, & otherwise \end{cases} \tag{22}$$

where x indicates the number of cores related to each service provider. The linguistic term of the parameter 'occupied buffer' is specified as follow

$$O(x) = \begin{cases} empty, & 0 \le x < 33 \\ partial, & 33 \le x < 66 \\ full, & otherwise \end{cases} \tag{23}$$

where x indicates the occupied buffer of each service provider. Note that the occupied buffer of any service provider is calculated as follow

$$O_B = \frac{N_R}{I_B} \times 100 \tag{24}$$

where N_R is the number of requests existed in the buffer and I_B is the initial buffer spaces of the service providers. The

Table 4 The initial knowledge bases and the total matrixes of the service providers in the BBDM method

Service provider no.	Behavioral style	d_s	v_s	γ	Behavioral matrix	Total matrix
1	Slow	0.33	0.2	0.8	$\begin{bmatrix} 0.33 & 0.200 & 0.160 \\ 0.33 & 0.200 & 0.160 \\ 0.34 & 0.201 & 0.161 \end{bmatrix}$	$\begin{bmatrix} 0.33 & 0.1249 & 0.1600 \\ 0.33 & 0.1249 & 0.1600 \\ 0.34 & 0.1255 & 0.1607 \end{bmatrix}$
	Light	0.33	0.3	1	$\begin{bmatrix} 0.33 & 0.300 & 0.300 \\ 0.33 & 0.300 & 0.300 \\ 0.34 & 0.301 & 0.301 \end{bmatrix}$	
	Poor	0.34	0.1	0.2	$\begin{bmatrix} 0.33 & 0.099 & 0.020 \\ 0.33 & 0.099 & 0.020 \\ 0.34 & 0.100 & 0.020 \end{bmatrix}$	
2	Heavy	0.33	1	0.5	$\begin{bmatrix} 0.33 & 1.000 & 0.500 \\ 0.33 & 1.000 & 0.500 \\ 0.34 & 1.000 & 0.500 \end{bmatrix}$	$\begin{bmatrix} 0.33 & 0.7478 & 0.6030 \\ 0.33 & 0.7478 & 0.6030 \\ 0.34 & 0.7481 & 0.6033 \end{bmatrix}$
	Strong	0.34	0.9	0.7	$\begin{bmatrix} 0.33 & 0.899 & 0.629 \\ 0.33 & 0.899 & 0.629 \\ 0.34 & 0.900 & 0.630 \end{bmatrix}$	
	Slow	0.33	0.2	0.1	$\begin{bmatrix} 0.33 & 0.200 & 0.020 \\ 0.33 & 0.200 & 0.020 \\ 0.34 & 0.201 & 0.020 \end{bmatrix}$	
	Strong	0.34	0.9	1	$\begin{bmatrix} 0.33 & 0.899 & 0.899 \\ 0.33 & 0.899 & 0.899 \\ 0.34 & 0.900 & 0.900 \end{bmatrix}$	
	Fast	0.33	1	0.9	$\begin{bmatrix} 0.33 & 1.000 & 0.900 \\ 0.33 & 1.000 & 0.900 \\ 0.34 & 1.000 & 0.900 \end{bmatrix}$	
3	Poor	0.34	0.1	0.75	$\begin{bmatrix} 0.33 & 0.099 & 0.074 \\ 0.33 & 0.099 & 0.074 \\ 0.34 & 0.100 & 0.075 \end{bmatrix}$	$\begin{bmatrix} 0.33 & 0.6028 & 0.5099 \\ 0.33 & 0.6028 & 0.5099 \\ 0.34 & 0.6031 & 0.5106 \end{bmatrix}$
	Fast	0.33	1	0.95	$\begin{bmatrix} 0.33 & 1.000 & 0.950 \\ 0.33 & 1.000 & 0.950 \\ 0.34 & 1.000 & 0.950 \end{bmatrix}$	
	Strong	0.34	0.9	1	$\begin{bmatrix} 0.33 & 0.899 & 0.899 \\ 0.33 & 0.899 & 0.899 \\ 0.34 & 0.900 & 0.900 \end{bmatrix}$	
	Heavy	0.33	1	0.5	$\begin{bmatrix} 0.33 & 1.000 & 0.500 \\ 0.33 & 1.000 & 0.500 \\ 0.34 & 1.000 & 0.500 \end{bmatrix}$	
4	Strong	0.34	0.9	0.8	$\begin{bmatrix} 0.33 & 0.899 & 0.719 \\ 0.33 & 0.899 & 0.719 \\ 0.34 & 0.900 & 0.720 \end{bmatrix}$	$\begin{bmatrix} 0.33 & 0.7096 & 0.7144 \\ 0.33 & 0.7096 & 0.7144 \\ 0.34 & 0.7100 & 0.7150 \end{bmatrix}$
	Heavy	0.33	1	0.7	$\begin{bmatrix} 0.33 & 1.000 & 0.700 \\ 0.33 & 1.000 & 0.700 \\ 0.34 & 1.000 & 0.700 \end{bmatrix}$	
5	Light	0.33	0.3	1	$\begin{bmatrix} 0.33 & 0.300 & 0.300 \\ 0.33 & 0.300 & 0.300 \\ 0.34 & 0.301 & 0.301 \end{bmatrix}$	$\begin{bmatrix} 0.33 & 0.3000 & 0.3000 \\ 0.33 & 0.3000 & 0.3000 \\ 0.34 & 0.3010 & 0.3010 \end{bmatrix}$

linguistic term of the parameter 'service rate' is defined as follow

$$S(x) = \begin{cases} low, & 0 \le x < 0.3 \\ medium, & 0.3 \le x < 0.6 \\ high, & otherwise \end{cases} \tag{25}$$

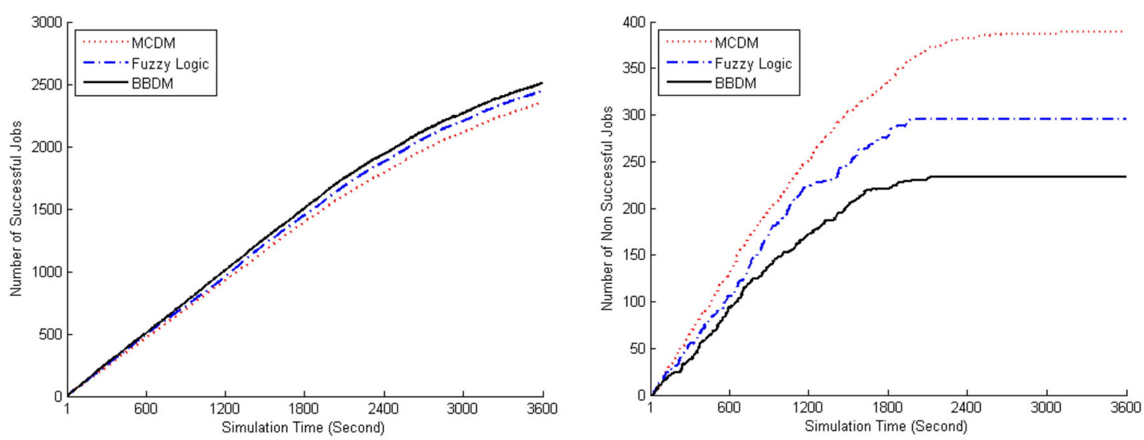
where x indicates the service rate of each service provider. Note that the service rate of any service provider is calculated as follow

$$S_R = \frac{N_C}{N_R} \tag{26}$$

where $\mathrm{N_C}$ is the number of completed requests by the service provider and $\mathrm{N_R}$ is the number of requests delivered

Table 5 The rules used in the MCDM and fuzzy logic methods

Rule no.	Antecedent			Consequent
	Number of cores	Occupied buffer	Service rate	Success probability
1	Few	Empty	Low	Unsuccessful
2	Few	Empty	Medium	Moderate
3	Few	Empty	High	Successful
4	Few	Partial	Low	Unsuccessful
5	Few	Partial	Medium	Unsuccessful
6	Few	Partial	High	Moderate
7	Few	Full	Low	Unsuccessful
8	Few	Full	Medium	Unsuccessful
9	Few	Full	High	Moderate
10	Mediocre	Empty	Low	Unsuccessful
11	Mediocre	Empty	Medium	Moderate
12	Mediocre	Empty	High	Moderate
13	Mediocre	Partial	Low	Unsuccessful
14	Mediocre	Partial	Medium	Moderate
15	Mediocre	Partial	High	Moderate
16	Mediocre	Full	Low	Unsuccessful
17	Mediocre	Full	Medium	Unsuccessful
18	Mediocre	Full	High	Moderate
19	Many	Empty	Llow	Moderate
20	Many	Empty	Medium	Successful
21	Many	Empty	High	Successful
22	Many	Partial	Low	Moderate
23	Many	Partial	Medium	Moderate
24	Many	Partial	High	Successful
25	Many	Full	Llow	Unsuccessful
26	Many	Full	Medium	Unsuccessful
27	Many	Full	High	Moderate



(a) (b)

Fig. 15 Simulation results of the resource discovery in a Cloud-based system by the simulated methods where the number of service providers is 5, the number of customers is 10, and the buffer space is 5. **a** Number of successful jobs, **b** number of non successful jobs

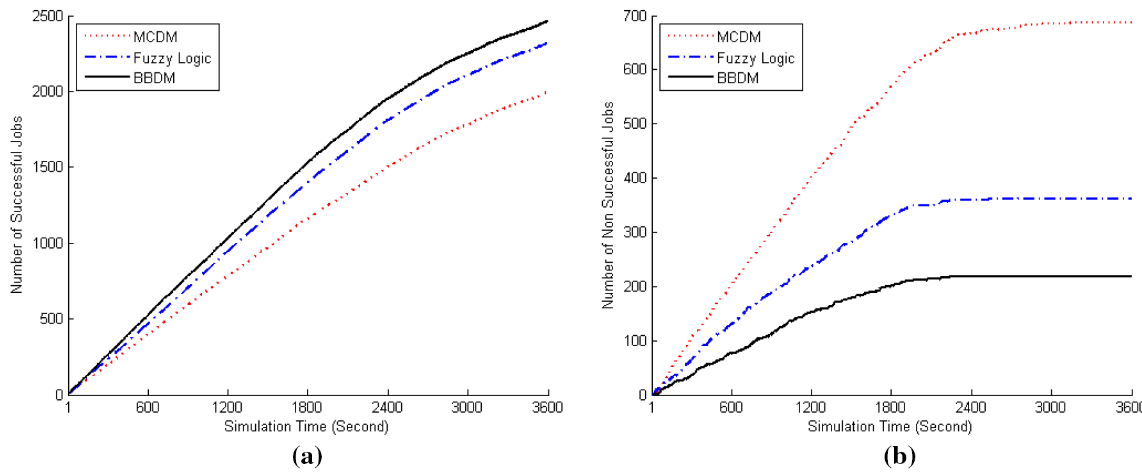


Fig. 16 Simulation results of the resource discovery in a Cloud-based system by the simulated methods where the number of service providers is 5, the number of customers is 20, and the buffer space is 5. **a** Number of successful jobs, **b** number of non successful jobs

to the service provider. The linguistic term of the parameter 'success probability' is determined as follow

$$SP(x) = \begin{cases} unsuccessful, & 0 \le x < 33 \\ moderate, & 33 \le x < 66 \\ successful, & otherwise \end{cases} \tag{27}$$

where x indicates the success probability of each service provider. In the simulated fuzzy logic method, membership functions of the parameter 'number of cores' are determined by triangular method [35] and membership functions of other parameters are determined by bell-shaped method [36]. The total fuzzy rule is defined by Mamdani fuzzy model [37] as well as any output fuzzy set is converted to a crisp value by center of gravity method [38]. Note that in the simulated MCDM and fuzzy logic methods, the Cloud service provider with the highest success probability is selected as a proper provider to serve the generated requests.

7.2 Comparison results

Simulation results of the MCDM, fuzzy logic, and BBDM methods to serve various requests of the customers are shown in Fig. 15; where the number of Cloud service providers is equal to 5, the number of customers is equal to 10, and the initial buffer spaces of the service providers are equal to 5. They demonstrate that the number of successful jobs obtained by the proposed method is more than that obtained by the MCDM and fuzzy logic methods. Moreover, the number of non successful jobs in the proposed method is less than that in the other simulated methods. As depicted in the comparison results, the quantitative difference between the number of successful jobs in the BBDM method and the other simulated methods is not very considerable. In contrast, the quantitative difference between the number of non successful jobs in the proposed method and the other methods is very noticeable after passing 800 seconds.

Simulation results of the MCDM, fuzzy logic, and BBDM methods to complete various requests of the customers are illustrated by Fig. 16; where the number of Cloud service providers is equal to 5, the number of customers is equal to 20, and the initial buffer spaces of the service providers are equal to 5. Comparison results show that the proposed method surpasses the other simulated methods in terms of the number of successful jobs and the number of non successful jobs. They demonstrate that the service performance in the BBDM method is higher than that in the fuzzy method as well as the service performance in the BBDM method is very higher than that in the MCDM method. Therefore, the network efficiency obtained by the BBDM and fuzzy logic methods is more than that obtained by the MCDM method when the number of customers is equal to 20.

Simulation results of the MCDM, fuzzy logic, and BBDM methods to accomplish various requests of the customers are depicted in Fig. 17; where the number of Cloud service providers is equal to 5, the number of customers is equal to 30, and the initial buffer spaces of the service providers are equal to 10. They demonstrate that the efficiency of the proposed method is better than the other simulated methods in aspects of the number of successful jobs and the number of non successful jobs. Furthermore, the results of the MCDM and fuzzy logic methods are near to each other when the number of customers is equal to 30 and the initial buffer space is equal to 10 requests. Comparison results show that the network performance achieved by the proposed method is, undoubtedly, higher than that achieved by the other methods under various network conditions. However, the number of non successful jobs obtained by the MCDM and fuzzy logic methods is very higher than that obtained by the BBDM method in most of the simulation times.

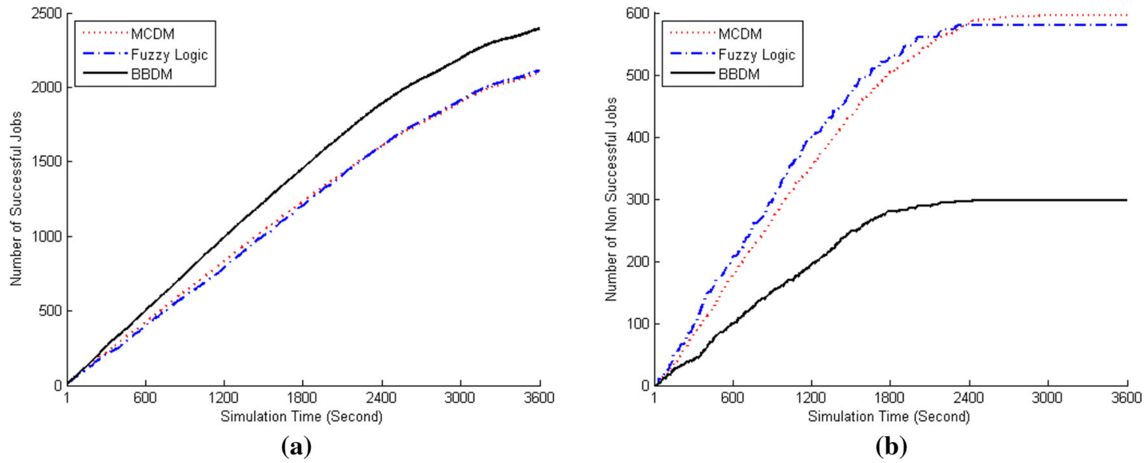


Fig. 17 Simulation results of the resource discovery in a Cloud-based system by the simulated methods where the number of service providers is 5, the number of customers is 30, and the buffer space is 10. **a** Number of successful jobs; **b** number of non successful jobs

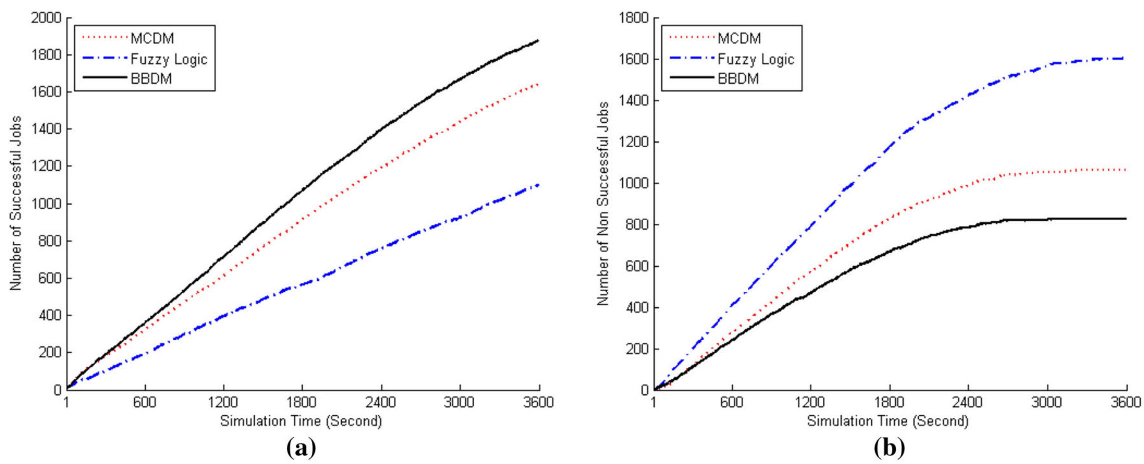


Fig. 18 Simulation results of the resource discovery in a Cloud-based system by the simulated methods where the number of service providers is 5, the number of customers is 50, and the buffer space is 20. **a** Number of successful jobs, **b** number of non successful jobs

Simulation results of the MCDM, fuzzy logic, and BBDM methods to serve various requests of the customers are shown in Fig. 18; where the number of Cloud service providers is equal to 5, the number of customers is equal to 50, and the initial buffer spaces of the service providers are equal to 20. Similar to the previous comparison results, the efficiency of the proposed method is better than the other simulated methods. However, the performance of the MCDM method is very higher than the fuzzy logic method when the number of customers and the initial buffer spaces of the service providers increase, considerably. The quantitative difference between the number of successful jobs in the BBDM method and the other simulated methods is very obvious on whole the simulation times. Furthermore, the quantitative difference between the number of non successful jobs in the proposed method and the other methods is very clear after the half-time of the simulation process.

8 Conclusion

In this paper, behavior-based decision making, called BBDM, is proposed to use humanistic behavioral reasoning for appropriate decision making in control and system engineering. It is a knowledge-based method having the learning ability to solve some of the existing decision problems. The knowledge base includes prior behavioral styles (e.g., the accurate in the driving behavior) to make proper decisions when any desired behavioral style is requested from the system. All of the decision-based operations in the BBDM method are performed by the proposed behavioral decision system, called BDS. When a desired behavioral style is requested from the system, a success rate is calculated by BDS based on the obtained knowledge base to forecast the success probability of the requested style. BDS is composed of three main units: decomposition, behavioral inference, and composition. The decomposition unit converts any behavioral style to the behavioral set which consists of multiple optional features (e.g., the speed and the accuracy in the driving behavior). Any feature has two attributes: importance factor and belongingness factor. Importance factor indicates the degree of the feature and belongingness factor indicates the value of the feature. These attributes can be specified by manual way or one of the proposed decomposition methods. In the behavioral inference unit, the information of any behavioral set is stored in the behavioral matrix which consists of three attributes 'degree', 'value' and 'truth'. After behavioral matrixes of the initial behavioral styles are created by the system, the total matrix is produced by one of the proposed behavioral aggregation methods. The total matrix represents the total behavior of the system that includes three attributes 'degree', 'truth' and 'growth'. When a desired behavioral style is requested from the system, a success set is produced based on the desired style and the obtained total matrix. Because the real world deals with real numbers, the composition unit converts any success set to a success rate by one of the proposed composition methods. Note that the proposed decision system has a learning engine to update the total behavior anytime via obtaining new behavioral information. The learning ability can solve any possible problems existed in the system.

Functionality of the proposed BBDM method is compared to some of the existing intelligent methods in terms of decision making type, humanistic data reasoning, humanistic behavioral reasoning, public knowledge base, private knowledge base, reasoning type, decision making for systems, and decision making for humans. Furthermore, the BBDM method is compared to multi-criteria decision making (MCDM) and fuzzy logic methods to serve various requests of the customers in a Cloud-based system. Simulation results demonstrate that the number of successful jobs achieved by the BBDM method could be increased by nearly 30% more than that obtained by the fuzzy logic method and by nearly 20% more than that obtained by the MCDM method. Furthermore, the number of non successful jobs achieved by the BBDM method could be decreased by nearly 45% less than that obtained by the fuzzy logic method and by nearly 50% less than that obtained by the MCDM method.

The BBDM method is proposed to be utilized by various applications and control systems to make appropriate decisions under numerous environmental conditions. It presents a novel behavior-based decision system to science world in order to solve some of the existing decision problems. Therefore, new decomposition, aggregation, inference, and composition methods could be presented by other researchers in the future.

References

1. Juuso EK (2004) Integration of intelligent systems in development of smart adaptive systems. Int J Approx Reason 35(3):307–337
2. Hatzilygeroudis I, Prentzas J (2011) Combinations of intelligent methods and applications. Springer, Berlin
3. Kaparulin DS, Lyakhovich SL, Sharapov AA (2013) BRST analysis of general mechanical systems. J Geom Phys 74:164–184
4. Lavopa E, Zanchetta P, Sumner M, Cupertino F (2009) Real-time estimation of fundamental frequency and harmonics for active shunt power filters in aircraft electrical systems. IEEE Trans Industr Electron 56(8):2875–2884
5. Flynn MJ, Luk W (2011) Computer system design: system-on-chip. Wiley, Hoboken
6. Cassidy AS, Georgiou J, Andreou AG (2013) Design of silicon brains in the nano-CMOS era: spiking neurons, learning synapses and neural architecture optimization. Neural Netw 45:4–26
7. Bueno-Crespo A, García-Laencina PJ, Sancho-Gómez J-L (2013) Neural architecture design based on extreme learning machine. Neural Netw 48:19–24
8. Huang G-B, Zhu Q-Y, Siew C-K (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1–3):489–501
9. Alpaydin E (2014) Introduction to machine learning. MIT Press, Cambridge
10. Ho W, Xu X, Dey PK (2010) Multi-criteria decision making approaches for supplier evaluation and selection: a literature review. Eur J Oper Res 202(1):16–24
11. Mela K, Tiainen T, Heinisuo M (2012) Comparative study of multiple criteria decision making methods for building design. Adv Eng Inform 26(4):716–726
12. Harris J (2001) An introduction to fuzzy logic applications, vol 23. Springer, Berlin
13. Azadegan A, Porobic L, Ghazinoory S, Samouei P, Kheirkhah AS (2011) Fuzzy logic in manufacturing: a review of literature and a specialized application. Int J Prod Econ 132(2):258–270
14. Fragiadakis NG, Tsoukalas VD, Papazoglou VJ (2014) An adaptive neuro-fuzzy inference system (anfis) model for assessing occupational risk in the shipbuilding industry. Saf Sci 63:226–235
15. Hadjileontiadou SJ, Dias SB, Diniz JA, Hadjileontiadis LJ (2015) Fuzzy logic-based modeling in collaborative and blended learning. IGI global in advances in educational technologies and instructional design (AETID) series
16. Eiben AE, Schoenauer M (2002) Evolutionary computing. Inf Process Lett 82(1):1–6
17. Yang S, Li M, Liu X, Zheng J (2013) A grid-based evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 17(5):721–736
18. Hung D (2001) Theories of learning and computer-mediated instructional technologies. Educ Media Int 38(4):281–287
19. Durand G, Belacel N, LaPlante F (2013) Graph theory based model for learning path recommendation. Inf Sci 251:10–21
20. Schmidhuber J (2015) Deep learning in neural networks: an overview. Neural Netw 61:85–117
21. Yurish SY (2010) Smart sensors systems design: new curricula based on marie curie chairs excellence (EXC) project's results. Measurement 43(1):21–30
22. Lei C-U, Wan K, Man KL (2013) Developing a smart learning environment in universities via cyber-physical systems. Procedia Comput Sci 17:583–585
23. Gintis H, Bowles S, Boyd R, Fehr E (2003) Explaining altruistic behavior in humans. Evolut Hum Behav 24(3):153–172
24. Zadeh LA (1965) Fuzzy sets. Inf Control 8(3):338–353
25. Almond RG, Mislevy RJ, Steinberg LS, Yan D, Williamson DM (2015) Bayesian networks in educational assessment. Springer, Berlin

26. Cai B, Huang L, Xie M (2017) Bayesian networks in fault diagnosis. IEEE Trans Industr Inf 13(5):2227–2240
27. Soualhi A, Razik H, Clerc G, Doan DD (2014) Prognosis of bearing failures using hidden Markov models and the adaptive neuro-fuzzy inference system. IEEE Trans Industr Electron 61(6):2864–2874
28. Suparta W, Alhasa KM (2016) Adaptive neuro-fuzzy interference system. In: Modeling of tropospheric delays using ANFIS. Springer, pp 5–18
29. Robb A, White C, Cordar A, Wendling A, Lampotang S, Lok B (2015) A comparison of speaking up behavior during conflict with real and virtual humans. Comput Hum Behav 52:12–21
30. Ristea I (2013) Reflections on mechanisms influencing human behavior. Procedia-Soc Behav Sci 92:799–805
31. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I (2009) Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. Future Gener Comput Syst 25(6):599–616
32. Marston S, Li Z, Bandyopadhyay S, Zhang J, Ghalsasi A (2011) Cloud computing–the business perspective. Decis Support Syst 51(1):176–189
33. Gharajeh MS (2015) The significant concepts of cloud computing: technology, architecture, applications, and security. CreateSpace Independent Publishing Platform
34. Gupta RD, Kundu D (2001) Generalized exponential distribution: different method of estimations. J Stat Comput Simul 69(4):315–337
35. Pedrycz W (1994) Why triangular membership functions? Fuzzy Sets Syst 64(1):21–30
36. Zhao J, Bose BK (2002) Evaluation of membership functions for fuzzy logic controlled induction motor drive. In: IEEE 28th annual conference of the industrial electronics society (IECON), 5–8, vol 1, pp 229–234
37. Mamdani EH, Assilian S (1975) An experiment in linguistic synthesis with a fuzzy logic controller. Int J Man Mach Stud 7(1):1–13
38. Ross TJ (2004) Fuzzy logic with engineering applications, 2nd edn. Wiley, Hoboken