A Learning-Embedded Attributed Petri Net to Optimize Student Learning in a Serious Game

Jing Liang, Ying Tang, *Senior Member, IEEE*, Ryan Hare, *Member, IEEE*, Ben Wu, *Member, IEEE*, and Fei-Yue Wang, *Fellow, IEEE*

***Abstract*— Serious games (SGs) are a practice of growing importance due to their high potential as an educational tool for augmented learning. However, little effort has been devoted to address student learning optimization in an SG from a systematic point of view. This article tackles this challenge by developing a learning-embedded attribute Petri net (LAPN) model to represent game flow and student learning decision-makings. The dynamics of learner behaviors in game are then addressed through the incorporation of learning mechanisms (i.e., reinforcement learning (RL) and random forest classification) into the Petri net model for knowledge reasoning and learning. Finally, an algorithm based on LAPN is proposed, aiming to guide learners to achieve a faster and better solution to problem-solving in game. The benefit of the proposed model and algorithm is then demonstrated in the SG *Gridlock*.**

***Index Terms*— Learning optimization, Petri nets (PNs), serious game (SG).**

I. Introduction

CONSIDERABLE interest has been devoted to the pursuit of learning and training both through and with digital games. The so-called serious games (SGs) are "computer-based games with a primary purpose other than entertainment" [1]. Compared to traditional textbook-driven lecturing, educational SGs offer many more benefits than just visualization. These games are interactions within immersive virtual worlds that promote learning and problem-solving through authentic and engaging play [2]. Experiential consequentiality and various rewards embedded in them motivate players to develop their skills at their own pace in a nonjudgmental but competitive and fun environment [3], [4].

A perusal of current literature reveals that although SGs contain pedagogical aspects, most of them are based on unguided discovery [5]. While such minimal guidance might work for self-motivated players, it can be very frustrating for those who lack motivation and prior knowledge, resulting in memory overwork and decreased learning [6]. A number of studies have undertaken this issue and explored the ways of providing scaffolding in games without disrupting engagement and fun. One group examines various ways to embed assessment directly into games [7]–[10]. The purpose of this embedded assessment is to collect gameplay information and infer players' capabilities and degrees of knowledge. A second, similar line of work focuses on the support in games for learners' cognitive development [11]–[13]. The guidance is often designed generically to fit all players. From our prior study [14], we found that such standardized guidance does not always meet individual player's needs. Understanding personal differences through gameplay behaviors and offering guidance tailored directly to their needs is emerging as a predominant issue in the current educational SG development.

To model a player, it is necessary to predict their cognitive, affective, and behavioral patterns. This modeling process has been an important area of research in the game community. Two recent and quite comprehensive reviews of the relevant literature are provided in [15] and [16]. For the purpose of our investigation, it is of interest to note the following from all the works discussed in [15] and [16]: 1) while a majority of model-based (top-down) approaches rely on social science theories to capture students' cognitive development, few of them attempt to model and optimize the sequence of learning in games and 2) while data-driven approaches using artificial intelligence and data mining techniques have gained much attention for player modeling in educational games, there has yet to be any effort to break deeper learning processes down into simpler and discrete mechanisms in games. In fact, educational games can be considered structured virtual environments where a sequence of learning is choreographed with tasks embodying units of domain-specific knowledge to be acquired by players. Therefore, a hybrid method, which takes advantage of both a formal model to characterize the in-game learning structure and a data-driven approach to explore a potentially large space of player interactions, is desirable to maximize learning.

Manuscript received May 31, 2021; revised November 19, 2021; accepted November 29, 2021. This work was supported in part by the National Science Foundation under Grant 1913809 and Grant 2121277. *(Corresponding author: Ying Tang.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Rowan's Institutional Review Board.

Jing Liang is with Squirrel AI Learning by Yixue Education Inc., Highland Park, NJ 08904 USA.

Ying Tang is with the Department of Electrical and Computer Engineering, Rowan University, Glassboro, NJ 08028 USA, and also with the Institute of Smart Education, Qingdao Academy of Intelligent Industries, Qingdao 266000, China (e-mail: tang@rowan.edu).

Ryan Hare and Ben Wu are with the Department of Electrical and Computer Engineering, Rowan University, Glassboro, NJ 08028 USA.

Fei-Yue Wang is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the Institute of Smart Education, Qingdao Academy of Intelligent Industries, Qingdao 266000, China.

Digital Object Identifier 10.1109/TCSS.2021.3132355

Motivated by the above general remarks, this article proposes a systematic approach to address the dynamics of learner behaviors in educational games for learning optimization and makes contributions in two areas. First, we present a special class of Petri nets (PNs) called learning-embedded attributed Petri nets (LAPNs), which are designed to represent a learner's knowledge, reasoning, and learning process. A set of attributes are associated with tokens in LAPN to represent the personal differences of various learners. The well-formed formalism of PNs makes it practical to explicitly model all possible decisions that a specific player can take in game, along with the inputs and outputs of each action. Second, the incorporation of learning mechanisms (i.e., reinforcement learning (RL) and random forest (RF) classification) into the PN allows the system to automatically infer new knowledge and make better decisions, evolving over time. The rest of this article is organized as follows. The core game mechanics is presented in Section II. The LAPN that models the dynamic gameplay flow of a learner is illustrated in Section III, based on which the learning optimization process is conducted in Section IV followed by a case study in Section V. Finally, conclusions and perspectives are offered in Section VI.

II. Game Characterization

As pointed out in [17], many attempts have been made to harness the learning potential of games to deliver educational contents and intellectually engage students to solve problems. Such games often operate under the principle of four-step problem-solving: 1) problem comprehension and definition; 2) solution generation; 3) plan execution; and 4) solution revisit and improvement [18]. SGs designed with an ultimate problem-solving goal, therefore, can be partitioned into ω small tasks, each of which embodies a set of content knowledge that is part of the overall problem solution. Such a small task can be represented within the game environment by a mini-game, challenge, quiz, reading materials, or other educational or entertaining content. Regardless of a linear or nonlinear relationship of those tasks, completing a task not only provides learners with access to more tasks but also moves them one step closer to the final problem-solving goal. Consequently, as students play through this task-focused game, they must continually achieve higher understanding of the content to be able to progress.

As a learner plays, their performance can be easily measured through various metrics within the game environment. These responses to game stimuli can be quantized and recorded. For a player $\boldsymbol{X}_i$, their responses in the ω^{th} problem-solving task then form a feature vector $\upsilon_\omega(X_i) = (x_{i1}, x_{i2}, \ldots x_{iN})$, where $|\upsilon_\omega(X_i)| = N$ and N is the number of features considered in the ω^{th} problem-solving task that measures the performance of $\mathbf{X}_i$. A set of these vectors then serves as an input to the player model discussed later. In this article, we consider the following two types of player inputs as numerical elements of our student feature vectors.

A. In-Game Stimuli

Educational games often incorporate pedagogical strategies to elicit as many responses as possible from players during their gameplay. For instance, timely and context-sensitive question prompts have been proven as an important way of stimulating learners [19]. Speed-, accuracy-, or self-confidence-related measures of a player's answers to those prompts directly reflect his cognitive processes, which in turn can affect the player's experience or perception of the game world.

One of the main challenges in any game is teaching a player how to play. In educational games, the problem is further magnified by the high knowledge demand put on players when they are required to solve technical or educational problems. To address this, educational games often embed learning resources in addition to knowledge testing. Providing timely hints/support to guide a student through his knowledge acquisition is another viable method for this challenge.

B. External Stimuli

Games can also stimulate physiological changes within a player, which can be reflected on his facial expression [20], posture [21], gaze movement [22], and/or speech. These physiological indicators can be connected to the player's learning state (e.g., focused, distracted, and lost). Such measurements are meaningful to help understand a learner's response when their capacity to make decisions is compromised or sabotaged.

Combining all the aforementioned stimuli, educational games can be designed to make decisions on how to guide a player's exploration through the game world (e.g., which problem-solving tasks to tackle and in which order) and how to scaffold player's problem-solving processes (e.g., which knowledge to acquire and which help session to enter). To maximize student learning within such an educational game, these decisions are systematically modeled in Section III.

III. Learning-Embedded Attributed Petri Net

An SG can be viewed as a structured dynamic system where an individual player is placed in a continuous mode of interactions. As such, complex cognitive, affective, and behavioral responses are developed over time to evolve the game world, so to the player's experiences. With these characteristics in mind, this article extends a special class of Petri nets (PNs), i.e., called learning-embedded attributed Petri nets (LAPNs), to allow for a natural representation of players for in-game play decision-makings: 1) players with different feature vectors as attributes of tokens and 2) gameplay decisions as input–output mapping of transitions. In addition, two types of learning mechanisms are embedded with the model to account for learner dynamics in game, so the parameters of the model are adjusted by a data-driven approach through system interaction and evolution. For clarity, Section III-A gives a general definition of PNs. The adopted learning methods are then briefly presented in Section III-B, followed by a tailored LAPN definition in Section III-C.

A. Petri Nets

A PN is a graphical and mathematical tool for modeling, formal analysis, and design of discrete-event systems [23].

Pictorially, a PN is such a directed graph with places depicted by circles, transitions pictured by bars, and directed arcs connecting places to/from transitions. Each place may contain no tokens or any positive number of tokens pictured by small solid dots. As transitions within the PN are fired, the tokens "flow" through the net. With this flow, the PN can easily describe concurrent activities and keep track of limited resources or, for example, the position of a player within an SG in our case. Formally, a PN can be defined as follows.

Definition 1: A PN is defined as a five-tuple: PN $= (P, T, I, O, M)$.

1) $P = \{p_1, p_2, \ldots, p_n\}$ is a finite set of places that can be used to represent objects, resources, or status.
2) $T = \{t_1, t_2, \ldots, t_m\}$ is a finite set of transitions, $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$.
3) $I : P \times T \rightarrow \mathbb{N}$ is an input function that defines the set of directed arcs from P to T, where $\mathbb{N}$ is a nonnegative integer.
4) $O : P \times T \rightarrow \mathbb{N}$ is an output function that defines the set of directed arcs from T to P, where $\mathbb{N}$ is a nonnegative integer.
5) $M : P \rightarrow \mathbb{N}$ is a marking vector. $M(p)$ represents the number of tokens in the p place.

Extended from this traditional definition, various PNs have been proposed to allow for the construction of compact and parameterized models for different aspects of the modeled systems. For instance, time can be introduced to PNs and can be associated with places, transitions, and/or arcs to analyze the system performance [24], [25]. Predicate/transition nets have arcs to specify predicate [25], [26], and colored PNs allow tokens to carry values [25]. These extended PNs have seen applications in manufacturing [27]–[29] due to the ability of these nets to manage and track resources. They have even seen use in web service discovery [30], which works on human-centric data, similar to our approach that extends the traditional PN to model dynamic behaviors of learners in an SG.

B. Learning Mechanisms

The process of optimizing student learning in an educational SG can be modeled as a sequence of decisions selected from a set of feasible options. Starting from the initial game state, these options, in general, are contingent upon a player's current condition and their perceived learning needs. The goal of the system's decision-making is to advance the player one step closer to completing the game. From an operational standpoint, the derivation of the optimal sequence of player actions is straightforward in principle. However, much of the player information necessary for such decision-making is not readily available and can only be observed during the learning process, making for a challenging optimization problem.

Any attempt to account for player interactions with complex and uncertain environment must make use of a feedback mechanism that allows the system to monitor and assess players, must bring newly available "knowledge" into consideration for system decisions, and must be alert constantly to any possible changes in the underlying system environment.

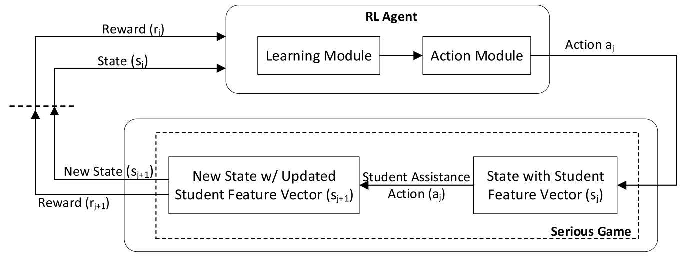

Fig. 1. Basic architecture of RL.

This approach is often characterized as online learning or data-driven approach. To that end, this article introduces two intelligent computing methods into the proposed PN framework to enable the learning capability of PN. With the augmentation, the proposed model not only provides a solid structure to manage the flow of play but also can learn and decide the action most appropriate for the player to take.

1) Random Forest: A random forest (RF) classifier is an ensemble of decision trees that provides an easy way to handle categorical, ordered, or numerical data. Each tree is grown from a subset of the original training data, where k features are randomly chosen from all N features in the training step ($k \ll N$). When training the tree, the Gini impurity [31] is used to select the best feature by which to split the training dataset at each node of the tree until the tree completely grows. Typically, the Gini impurity calculates the probability of a randomly chosen feature being incorrectly classified, so the training process focuses on minimizing this probability. Finally, for each data point sent into the RF, the category with the largest number of votes is selected as the result of classification. Considering that various types of data exist in player modeling, RF is used to classify the player's performance in terms of the efficiency and effectiveness of the player's solution to problem-solving tasks. We refer readers to [32] for the technical detail.

2) Reinforcement Learning: As shown in Fig. 1, reinforcement learning (RL) typically involves a learning controller called an agent that commands a sequence of actions to evolve the environment in a discrete state space [33]. The agent always follows a policy π and attempts to take the optimal action in the current state to maximize both immediate and future rewards defined as a state–action Q value function $Q(s, \mu)$ [34]

$$Q^{\pi}(s, \mu) = E^{\pi}\left[\sum_{k=0}^{\infty} \gamma^{k} r_{j+k+1} \middle| s_j = s, a_j = \mu, \pi\right] \quad (1)$$

where is r_j is a numerical reward given to the agent by the environment after taking action μ in state s, $\gamma \in [0, 1]$ is the discount factor that determines the weight of future rewards, and $E^{\pi}[\cdot]$ is the expected reward that the agent will obtain given that the agent follows policy π. Note that $\gamma \in [0, 1]$, meaning that the infinite sum in (1) would have a finite value as long as the reward is bounded. When $\gamma = 0$, the agent is nearsighted and only prioritizes immediate rewards. When $\gamma = 1$, the agent is farsighted and attempts to maximize all rewards. Typically, $0 < \gamma < 1$.

Here, the policy determines the probability of an agent selecting action μ when in state s that can be calculated using

Max–Boltzmann exploration [34]

$$\pi(s,\mu) = \frac{e^{\frac{Q(s,\mu)}{\Delta}}}{\sum_{j=1}^{n} e^{\frac{Q(s,\mu_j)}{\Delta}}} \tag{2}$$

where Δ is the temperature parameter that determines the uniformity of the probability spread, with higher values leading to a more uniform distribution. The value of Δ is dependent on the rewards given in a specific environment and can often be obtained through experimentation.

C. Learning-Embedded Attributed Petri Nets

Considering the heterogeneity of players' mastery of domain knowledge necessary for solving problems in game, attributes and their classifications are introduced into the previously defined PN model and associated with a particular set of places as defined in the following:

$$a(p) = \langle \cup_\omega \{v_\omega(p), \delta(v_\omega(p)\}\rangle \tag{3}$$

where $v_\omega(p)$ is the feature vector of the player represented by p in the ω^{th} game stage and $\delta(v_\omega(p))$ is the feature vector's class. For a player first coming into the game system, the attribute is initially set as $\langle\cdot\rangle$, where $\langle\cdot\rangle$ indicates that measurement and classification have yet to take place.

As stated in Section I, minimally guided instructional approaches are less effective and efficient. Desirable educational games should have mechanisms to not only elicit but also evaluate a player's experiences, based on which an optimal action that best fits the player's needs can be recommended for them. The evaluation and decision-making processes are modeled in LAPN as typical classification and action transitions, respectively.

The detailed definition of an LAPN is given in the following.

Definition 2: An LAPN is an extension of PN with seven-tuple: LAPN $= (P, T, I, O, M, A, \rho)$

1) P is a nonempty set, where $P = P^b \cup P^e$. The set of places in P^b represents player flow in game, and the set of places in P^e represents various game support.
   a) $P^b \cap P^e = \emptyset$.
   b) There is a unique $p \in P^b$ such that ${}^\bullet p = \emptyset$. This place is usually named root.
   c) $\forall p \in P$, $p^\bullet = \emptyset$, p is called a leaf.
2) T is partitioned into three subsets $T = T^g \cup T^h \cup T^l$. The set of transitions in T^g represents the classification based on the current player attribute, T^h player actions, and T^l player following various game support.
3) A: $P^b \to \langle U_\omega \{v_\omega(p), \delta(v_\omega(p)\}\rangle$ is a mapping function that maps p to a union of two-tuple. $\exists p \in P^b$ the root place, $a(p) = \langle\cdot\rangle$.
4) $\rho : T^h \cup T^l \to [0, 1]$ is a mapping function that maps t to a value, indicating the probability of selecting the action or game support transition t when the player at p, where $p \in {}^\bullet t \cap p \in P^b$.

As stated earlier, educational games often make use of various stimuli to elicit players' responses to ensure accurate player profiles in terms of knowledge mastery or learning preferences. To that end, the elicitation and RF classification are used and modeled as a special class of transitions T^g, whose firings change the attribute and its corresponding class of their outgoing places P^b.

In the context of learning optimization in educational games, each player's experiences also form an episode of states and state transitions in RL that can be directly modeled by P^b, T^h, and T^l in LAPN. As a result, a set of Q values calculated by (1) can be defined over the pairs $(p,\ t)$, where $p \in P^b$ and $t \in T^h \cup T^l$. Furthermore, the interpretation of the optimal Q value under a given policy in (2) is then directly connected to $\rho(t)$, recommending the player to always take the action assigned the highest probability and, by extension, the highest estimated improvement in their player profile (i.e., firing the transition t with the highest $\rho(t)$).

The transition enabling and firing rules are then defined in Definition 3.

Definition 3: The enabling and firing rules are as follows.

1) A transition $t \in T$ in LAPN is enabled if for all $p \in {}^\bullet t$
$$M(p) \ge I(p,t).$$
2) Firing an enabled $t \in T^g$ yields
$$\begin{aligned} M'(p) &= M(p) - I(p,t) + O(p,t) \\ &\quad \forall p \in {}^\bullet t \cup P^b, \exists q \in t^\bullet \cap P^b \\ a(q) &= \langle \cup_\omega \{\boldsymbol{Update}(v_\omega(p)), \\ &\quad \boldsymbol{RFClassify}(v_\omega(p))\}\rangle \end{aligned} \tag{4}$$
where **Update**() is the function to measure the player p feature vector and **RFClassify**() is the function of RF classification [32].
3) Firing an enabled $t \in T^h \cup T^l$ with the highest $\rho(t)$ yields
$$\begin{aligned} M'(p) &= M(p) - I(p,t) + O(p,t) \\ &\quad \forall p \in {}^\bullet t \cup P^b, \exists q \in t^\bullet \cap P^b \\ a(q) &= a(p). \end{aligned}$$

IV. Learning Optimization

Research has found that learning is often maximized when the learner is actively involved in the cognitive process of problem-solving and receives timely feedback to reflect and regulate this process. In LAPN, each player's involvement is tracked through the game flow, which is modeled as the token moving through a sequence of places in P^b. The player's responses to various stimuli are then measured at discrete times. As the player's in-game behavior directly impacts the individualized guidance recommended to them, the feature vectors are introduced to represent player aptitudes, as are the classifications via the RF. Thus, this combined set of information is designed to inform the system and steer the feedback given to the students.

At any point in time, the game system selects the best action from an action pool to the player according to his attribute at a certain game state. Such state–action pairs are modeled as the pairs $(p,\ t)$ in LAPN, where $p \in P^b$ and $t \in T^h \cup T^l$. A probability value is then introduced to each action/support transition and is updated via RL policy using (2). This policy is continually altered based on the player's attribute and some evaluative feedback that determines the numerical reward given to the system. This numerical reward, in this case, can be directly based off improvements (or declines) in the student's performance indicators. In other words, the game system continuously observes player behavior and iteratively learns the best action to take at any time.

Finally, the learning optimization algorithm is proposed with the aim to maximize the efficiency and effectiveness of a player's solution in terms of less iterations and errors. The process can be done by identifying the order of actions (i.e., action transition firing in LAPN) that maximizes the accumulated rewards. Based on the structure of the LAPN, the objective is then achieved by traversing the LAPN from the root place and selecting the action transition with the highest probability. The detail is given in Algorithm 1. Fig. 2 summarizes the overall logic of the approach.

Algorithm 1 LAPN Learning Optimization

1. **SET** $Z = \{p_0\}$, assuming p_0 is the root place for a player to start the game
2. **SET** TS $= \emptyset$ (TS - action/support transition set)
3. **WHILE** $(Z \neq \emptyset)$ **DO**:
4. **FOR** each node p in Z **DO:**
5. **SET** $\mathrm{H}(p) = \emptyset$ ($\mathrm{H}(p)$ - outgoing transition set of p)
6. **IF** $\forall t \in p^\bullet \cap t \in T^g$
7. Fire t, update the attribute and class of the player's token using Eq. 4, and deposit it into $t^\bullet$
8. **ELSE IF** $\forall t \in p^\bullet \cap t \in T^h \cup T^l$
9. $\mathrm{H}(p) = H(p) \cup \{t\}$
10. Update $\rho(t)$
11. Order transitions in $H(p)$ in a descending order of their $\rho(t)$
12. Select the first transaction t in $\mathrm{H}(p)$ to fire, and deposit token(s) with the attribute and class unchanged into $t^\bullet$
13. $Z = Z \cup \{t^\bullet \cap P^b\}$
14. $TS = TS \cup \{t\}$
15. $Z = Z - \{p\}$
16. **END**

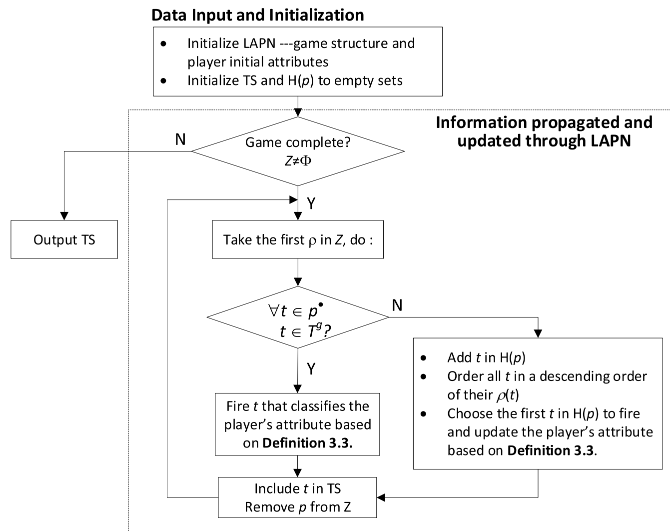

Fig. 2. Logic of player action planning.

V. Case Study

To better understand our method, this section presents a case study, where the LAPN model and associated optimization algorithm are applied to an existing SG called *Gridlock*. Section V-A briefly describes the game logic and its LAPN representation, whereas Section V-B details the results from a comparison test of *Gridlock* with and without learning optimization.

A. Gridlock Modeling

Gridlock is a first-person SG that positions a player in a four-way intersection to witness a serious traffic accident due to malfunctional traffic lights [32]. The player is then assigned to solve the problem using his knowledge, skills, and concepts learned in curricular content (for this case, sequential circuit design in digital logic design) to solve the problem. Doing this, transformational play is designed to spark his interest and lead to deeper engagement with the content presented.

Gridlock splits up the final design goal of a traffic light controller into eight problem-solving sections, each of which tasks students with specific concepts related to the overall goal, such as binary logic, finite state machine design, and Verilog code syntax. We refer these problem-solving sections as subject-specific learning blocks. To provide a guided learning and problem-solving experience, *Gridlock* integrates interventions throughout the game to elicit the following player responses.

1) *Question Prompts and Answers:* A set of questions on the related content knowledge is embedded in each game section. The player's answers to those questions are evaluated based on their correctness. Other metrics from prompts are also recorded, including time taken to answer and a student-provided confidence level.
2) *Emotion Values:* A facial emotion detection engine is integrated with the game to provide the classification of seven standard emotions: happiness, sadness, surprise, fear, disgust, anger, and neutral.
3) A frustration metric is measured and defined as a function of the player's sporadic rapid mouse movements or keyboard strokes over set periods of time.

The overall system can be visualized as a virtual game connected with an adaptation engine. Fig. 3 shows the system architecture of *Gridlock*. Within the game, the learning phase contains the content-specific problem-solving sections that the game directs students to. As the student plays, the adaptation engine collects data, feeding it to the RL decision-making component. This component computes transition probabilities for the LAPN, which tracks and directs the student to the various blocks and selects personalized assistance to provide within said blocks.

Fig. 4 shows the LAPN of *Gridlock*. Note that the portion of the LAPN for each subject-specific block is structurally identical. Therefore, not all blocks are shown in Fig. 4 for clarity. Blocks 1 and ω are presented in detail, while other blocks are

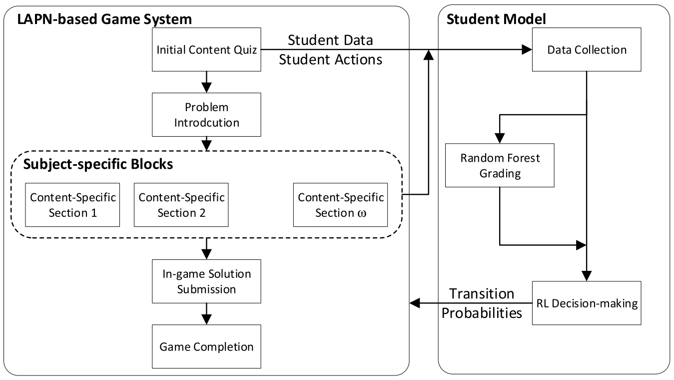

Fig. 3. Overall system architecture of *Gridlock*.

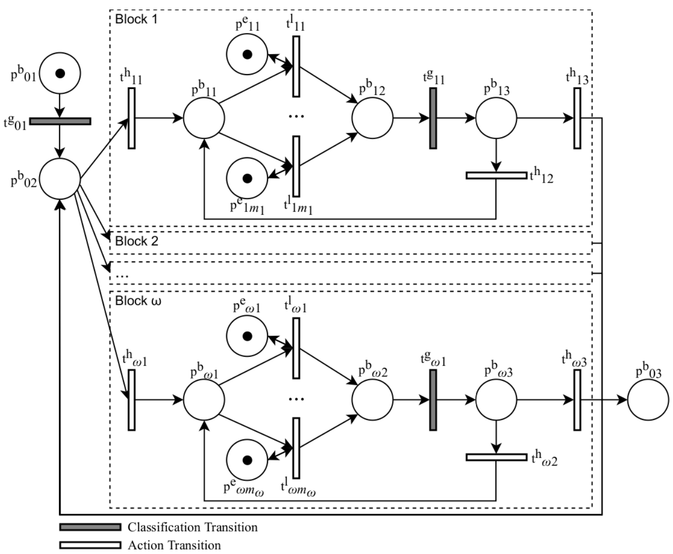

Fig. 4. LAPN of *Gridlock*.

omitted with dashed rectangles and ellipses representing their functional positions in the LAPN model.

A player/student is initialized in p^b_{01}, indicating their starting position in the game. A preliminary measure is then conducted and classified via t^g_{01}, based on which the player with the initial attributes and associated class in p^b_{02} is guided into one of the subject-specific blocks modeled as a set of transitions $\{t^h_{11}, t^h_{21}, \ldots, t^h_{n1}\}$, where $n = 8$ in *Gridlock*. In fact, the Q value defined over each pair (p^b_{02}, t^h_{y1}), $y = 1, 2, \ldots, n$ is calculated, which updates the corresponding probabilistic value $\rho(t^h_{y1})$. The transition with the highest ρ will be fired, leading the player into the learning block that can best improve his relevant knowledge.

In each subject-specific block (using Block 1 as the example), there is m_1 learning support $\{t^l_{11}, t^l_{12}, \ldots, t^l_{1m_1}\}$ for a player to choose. A similar Q-learning process is then carried out to guide him taking the best support. It is expected that the chosen support will improve the player's skill and knowledge level. Thus, an updated set of feature vectors is measured after the player takes the action and evaluated via t^g_{11}. Depending on improvements in the player's performance and mastery of the tasks given in the block, the player can be kept in a feedback loop (t^h_{12}) of repeated assistance until they demonstrate sufficient grasp of the concept, at which point they are returned to the main decision state, p^b_{02}.

To exemplify the system's decision-making process and demonstrate the performance of the proposed approach, a reduced version of *Gridlock* with three subject-specific learning blocks is considered in a case study where three types of measures are conducted for each subject, which are given in the following.

1) Average quiz score on the questions related to a specific concept in the range [0.0, 1.0] with 1.0 being a perfect score.
2) Total time taken to answer the related questions in a number of seconds.
3) Average confidence on their answers to the related questions in the range [1.0, 5.0] with 1 being not confident at all and 5 being very confident. In *Gridlock*, these values are obtained by a student self-rating system that asks students to rate their own confidence in their answer from not confident (1) to very confident (5).

When an average-performance-level student enters the game, Algorithm 1 starts its execution through the LAPN and the action/support sequence TS for the student is obtained. First, the token representing this player in p^b_{01} with $\langle\cdot\rangle$ attribute triggers the firing of t^g_{01}, leading a new token with the following attribute deposited into p^b_{02} via (4). Note that the classification values through the RF classifier are in the range [1, 3] with 1 being a bad score and 3 being a good score

$$a\big(p^b_{02}\big) = \langle\{[0.55, 41.2, 4.0], 1\}, \{[1.00, 18.4, 5.0], 3\}, \{[1.00, 48.7, 2.5], 2\}\rangle.$$

The above data show that the student had a low overall score and a longer than average time in the first block. They also indicated lower confidence in their answers, leading to a classification score 1 by the system. Meanwhile, the third block showed completely correct answers but still had a longer than average time. Furthermore, the student was not confident in their answers, leading to a classification score of 2.

Based on the attribute and classification, the system shows the Q values in the following. These values are learned by the system as it interacts with students, and they represent the total improvement expected for the given student when entering into each block

$$Q\big(a(p^b_{02}), t^h_{y1}\big) = \{8.216, -23.780, 7.739\}.$$

As shown, the system expects that firing transition 2 is expected to lead to a decrease in performance since the student has already demonstrated mastery of that content. Using those Q values and an experimentally determined temperature value $\Delta = 9$, the system updates the transition probabilities of t^h_{11}–t^h_{31} using (2)

$$\rho\big(t^h_{11}\big) = 0.621; \quad \rho\big(t^h_{21}\big) = 0.003; \quad \rho\big(t^h_{31}\big) = 0.377.$$

The system prioritizes the blocks in order and focuses on blocks the student shows low mastery in, skipping blocks that show higher levels of mastery. In this case, transition t^h_{11} is selected to fire and move the student into learning block 1, even though all transitions t^h_{11}–t^h_{31} are enabled. A new token with the same attribute and classification is then deposited into p^b_{11}, which is $a(p^b_{11}) = a(p^b_{02})$.

TABLE I
MANN–WHITNEY TEST COMPARISON

		N	Medium	Z	U	p
Comparison I	Baseline case	35	22	0.253	1364	0.400
	Simulated case		23.765			
Comparison II	Baseline case		22	3.766	808	$8.286 * 10^{-5}$
	Proposed case		13.998			

In learning block 1, there are $m_1 = 5$ possible help actions in this case. A similar process of calculating Q values and updating transition probabilities is conducted on transition t^l_{11}–t^l_{15} based on $a(p^b_{11})$ as shown in the following:

$$\rho(t^l_{11}) = 0.140;\quad \rho(t^l_{12}) = 0.087;\quad \rho(t^l_{13}) = 0.428$$
$$\rho(t^l_{14}) = 0.157;\quad \rho(t^l_{15}) = 0.188.$$

These probabilities for the help actions represent the estimated improvement the student will have from each of these actions based on the past experience. In this case, action 3 (t^l_{13}) is selected to help this student. His attribute is then remeasured and classified in t^g_{11} via (4)

$$a(p^b_{13}) = \langle\{[1.00, 34.4, 4.5], 3\}, \{[1.00, 18.4, 5.0], 3\}, \{[1.00, 48.7, 2.5], 2\}\rangle.$$

This process repeats a feedback loop of entering blocks, getting help, and updating attributes and classes until all three blocks show positive classifications that finally take the student to finish the game.

B. Comparison Results

The first set of experiments considers three cases. In the baseline case, a group of 53 students[1] played a standard version of *Gridlock* with the learning optimization schema disabled. In this case, students were free to explore *Gridlock* at their own pace and view content in any order they chose. They also played through all subject-specific blocks, regardless of their demonstrated proficiency. As such, students might play through sections of content that they had already mastered. Their recorded game performance is referred to here as the baseline case.

The players' responses to the game stimuli were measured at the beginning of the game as the initial feature vectors. The initial attributes of these 53 students were then input to a simulated case, where the system mimics how students play *Gridlock* with no guidance, referred to as the simulated case. The third case, so-called the proposed case, is an adjusted version of the simulated case with the same initial attributes of the 53 students as inputs but making use of the proposed learning optimization algorithm.

To show the difference between the three cases, the averaged time taken to complete each of the subject-specific blocks is measured and accumulated to calculate the overall time for game completion. Fig. 5 shows the comparison of the average game completion time between the three cases. The simulated case provides an accurate estimate of the baseline case, showing that the system can indeed simulate

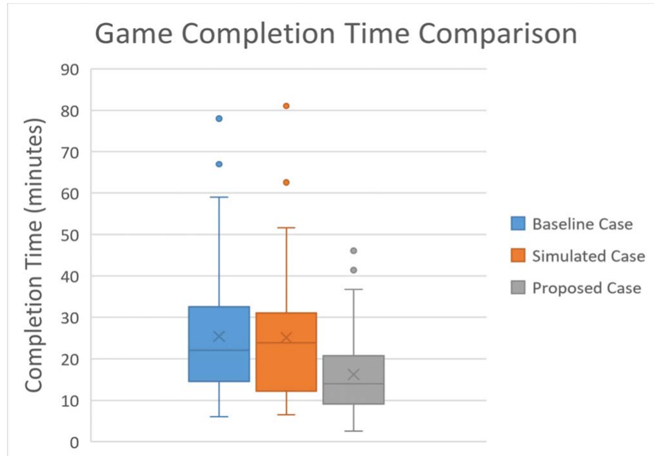

Fig. 5. Comparison of game completion time between the baseline and the proposed cases.

student performance. The proposed case, meanwhile, shows an improvement over the simulated case, indicating an effective proof-of-concept for the proposed system.

In addition, the baseline case is compared with the simulated and the proposed one, respectively. Mann–Whitney tests are used to measure their differences. As shown in Table I, the simulated case offers a statistically good approximation of students playing *Gridlock* in reality as there is no significant difference in student game completion time between the baseline and simulated cases ($p > 0.05$). On the contrary, a statistically significant difference is found when comparing the proposed case with the baseline one ($p \ll 0.05$). The proposed method guides a player in a way that focuses better on the content the player has issues with while skipping over the content that he has already mastered. In doing so, the game focuses on improving the player's areas of difficulty, so they can learn to solve the problem quickly and efficiently. Overall, players in the proposed case take less attempts to solve problems with less mistakes, resulting in a faster and better game experience.

To show the effectiveness of the system on different levels of students, a subset of 40 students is then split into groups based on their classroom performance as indicated by their grade point average (GPA). The average game completion time is compared between the baseline and proposed cases, as shown in Table II. Using these data, a 95% confidence interval for the true mean difference in average game completion time is calculated (the value of $g_{\alpha/2,b} = g_{0.0025,3} = 2.353$ from a t-distribution table) as follows:

$$7.115 \pm 2.353 * \frac{1.942}{\sqrt{4}}$$

or

$$4.830 \le \theta_1 - \theta_2 \le 9.400.$$

[1]The study has been approved by the Rowan University IRB Board since 2019.

TABLE II
COMPARISON OF AVERAGE GAME COMPLETION TIME FOR STUDENTS AT DIFFERENT LEVELS

GPA Range	Student Number	Avg. Time (minutes)		Observed Difference (minutes)
		Baseline case	Proposed case	
> 3.75	15	22.533	15.664	6.870
$3.5 - 3.74$	6	29.333	18.921	10.412
$3.0 - 3.49$	15	30.300	19.057	11.243
$2.5 - 2.99$	4	33.000	25.885	7.115
Mean		28.792	19.882	8.910
Standard Deviation				1.942

This 95% confidence interval lies exclusively above zero, providing strong evidence that the students with personalized guidance in *Gridlock* outperform the ones without any support in game because the average game completion time is significantly smaller statistically.

VI. CONCLUSION

SGs are emerging as an effective way of motivating and supporting learning. Although games are capable of tracking players' behavior, there are a few comprehensive studies in exploring the relation of game states and learning processes to maximize learning. To address this deficiency, this article proposes an LAPN model based on PNs, RL, and RF classification. Student learning is then represented as discrete intervention states in LAPN where individual player's performance is measured and characterized as attributes. Learner dynamics in game are tackled through the embedded learning mechanisms, so the parameters of the model are updated online as the system evolves. Finally, the learning optimization algorithm is derived based on LAPN to guide the player's decision-making at any given time, aiming to offer a faster and better solution to each problem-solving task in game. From the case study, the approach is found to be effective in decreasing game completion time.

The research can be extended in several directions. The proposed model and algorithm are designed to be easily adopted for any SGs. When the number of game stimuli increases, the size of the state space grows exponentially with each additional feature added. Methods to deal with high-dimensional state exploration deserve research efforts. Appling our approach to other SGs is needed to further test its validity and practicality for continuous improvement.

REFERENCES

[1] D. Djaouti, J. Alvarez, and J. P. Jessel, "Classifying serious games: The G/P/S model," in *Handbook of Research on Improving Learning and Motivation Through Educational Games: Multidisciplinary Approaches*. Hershey, PA, USA: IGI Global, 2011.

[2] C. Franzwa, Y. Tang, A. Johnson, and T. Bielefeldt, "Balancing fun and learning in a serious game design," *Int. J. Game-Based Learn.*, vol. 4, no. 4, pp. 37–57, Oct. 2014.

[3] Y. Tang *et al.*, "Sustain city: Effective serious game design in promoting science and engineering education," in *Design, Motivation, and Frameworks in Game-Based Learning*. Hershey, PA, USA: IGI Global, 2017.

[4] T. Terzidou, T. Tsiatsos, C. Miliou, and A. Sourvinou, "Agent supported serious game environment," *IEEE Trans. Learn. Technol.*, vol. 9, no. 3, pp. 217–230, Jul. 2016.

[5] F. Bellotti *et al.*, "Designing serious games for education: From pedagogical principles to game mechanisms," in *Proc. 5th Eur. Conf. Game-Based Learn.*, Athens, Greece, Oct. 2011, pp. 26–34.

[6] P. A. Kirschner, J. Sweller, and R. E. Clark, "Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching," *Educ. Psychol.*, vol. 41, no. 2, pp. 75–86, 2006.

[7] V. J. Shute, L. Wang, S. Greiff, W. Zhao, and G. Moore, "Measuring problem solving skills via stealth assessment in an engaging video game," *Comput. Hum. Behav.*, vol. 63, pp. 106–117, Oct. 2016.

[8] V. J. Shute, J. P. Leighton, E. E. Jang, and M.-W. Chu, "Advances in the science of assessment," *Educ. Assessment*, vol. 21, no. 1, pp. 1–27, 2016.

[9] L. Derbaliand and C. Frasson, "Players' motivation and EEG waves patterns in a serious game environment," *Intell. Tutoring Syst.*, vol. 6095, no. 2, pp. 297–299, 2010.

[10] M. A. Syufagi, M. Hariadi, and M. H. Purnomo, "Model of mental effort assessment in pedagogic games based on LVQ method," in *Proc. SESINDO Conf.*, Surabaya, Indonesia, Dec. 2008, pp. 556–564.

[11] F. Bellotti, B. Kapralos, K. Lee, P. Moreno-Ger, and R. Berta, "Assessment in and of serious games: An overview," *Adv. Hum.-Comput. Interact.*, vol. 2013, p. 1, Jan. 2013.

[12] C. S. Loh, "Information trails: In-process assessment of game-based learning," in *Assessment in Game-Based Learning* (Foundations, Innovations, and Perspectives). New York, NY, USA: Springer, 2012, pp. 123–144.

[13] B. Kim, H. Park, and Y. Baek, "Not just fun, but serious strategies: Using meta-cognitive strategies in game-based learning," *Comput. Educ.*, vol. 52, no. 4, pp. 800–810, May 2009.

[14] Y. Tang, J. Liang, R. Hare, and F.-Y. Wang, "A personalized learning system for parallel intelligent education," *IEEE Trans. Comput. Social Syst.*, vol. 7, no. 2, pp. 352–361, Apr. 2020.

[15] D. Hooshyar, M. Yousefi, and H. Lim, "A systematic review of data-driven approaches in player modeling of educational games," *Artif. Intell. Rev.* vol. 52, no. 3, pp. 1997–2017, 2019, doi: 10.1007/s10462-017-9609-8.

[16] A. M. Smith, C. Lewis, K. Hullet, and A. Sullivan, "An inclusive view of player modeling," in *Proc. 6th Int. Conf. Found. Digit. Games (FDG)*, May 2011.

[17] T. E. Coleman and A. G. Money, "Student-centred digital game–based learning: A conceptual framework and survey of the state of the art," *Higher Educ.*, vol. 79, no. 3, pp. 415–457, 2020.

[18] G. Polya, *How to Solve It: A New Aspect of Mathematical Method*, 2nd ed. Princeton, NJ, USA: Princeton Univ. Press, 1957.

[19] X. Ge, C.-H. Chen, and K. A. Davis, "Scaffolding novice instructional designers' problem-solving processes using question prompts in a web-based learning environment," *J. Educ. Comput. Res.*, vol. 33, no. 2, pp. 219–248, Sep. 2005.

[20] M. Pantic and G. Caridakis, "Image and video processing for affective applications," in *Emotion-Oriented Systems: The Humaine Handbook*. Berlin, Germany: Springer-Verlag, 2011, pp. 101–117.

[21] N. Savva, A. Scarinzi, and N. Bianchi-Berthouze, "Continuous recognition of player's affective body expression as dynamic quality of aesthetic experience," *IEEE Trans. Comput. Intell. AI in Games*, vol. 4, no. 3, pp. 199–212, Sep. 2012.

[22] J. Antunes and P. Santana, "A study on the use of eye tracking to adapt gameplay and procedural content generation in first-person shooter games," *Multimodal Technol. Interact.*, vol. 2, no. 2, p. 23, May 2018.

[23] R. Zurawski and M. Zhou, "Petri nets and industrial applications: A tutorial," *IEEE Trans. Ind. Electron.*, vol. 41, no. 6, pp. 567–583, Dec. 1994.

[24] Y. Tang, M. C. Zhou, and R. Caudill, "An integrated approach to disassembly planning and demanufacturing operation," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 773–784, Dec. 2001.

[25] J. Wang, *Timed Petri Nets: Theory and Application*. Norwell, MA, USA: Kluwer, 1998.

[26] Y. Deng, J. Wang, K. Beznosov, and J. Tsai, "An approach for modeling and analysis of security system architectures," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 5, pp. 1099–1119, Oct. 2003.

[27] J. Luo, Z. Liu, S. Wang, and K. Xing, "Robust deadlock avoidance policy for automated manufacturing system with multiple unreliable resources," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 3, pp. 812–821, May 2020.

[28] Z. Zhao, S. Liu, M. Zhou, D. You, and X. Guo, "Heuristic scheduling of batch production processes based on Petri nets and iterated greedy algorithms," *IEEE Trans. Autom. Sci. Eng.*, early access, Oct. 23, 2020, doi: 10.1109/TASE.2020.3027532.

[29] X. Guo, M. Zhou, S. Liu, and L. Qi, "Multiresource-constrained selective disassembly with maximal profit and minimal energy consumption," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 2, pp. 804–816, Apr. 2021.

[30] J. Sha, Y. Du, and L. Qi, "A user requirement oriented web service discovery approach based on logic and threshold Petri net," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 6, pp. 1528–1542, Nov. 2019.

[31] L. Rokach and O. Maimon, "Top-down induction of decision trees classifiers—A survey," *IEEE Trans. Syst., Man Cybern. C, Appl. Rev.*, vol. 35, no. 4, pp. 476–487, Nov. 2005, doi: 10.1109/TSMCC.2004.843247.

[32] R. Hare, Y. Tang, W. Cui, and J. Liang, "Optimize student learning via random forest-based adaptive narrative game," in *Proc. IEEE 16th Int. Conf. Autom. Sci. Eng. (CASE)*, Hong Kong, Aug. 2020, pp. 792–797.

[33] Q. Xiao, C. Li, Y. Tang, and L. Li, "Meta-reinforcement learning of machining parameters for energy-efficient process control of flexible turning operations," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 1, pp. 5–18, Jan. 2019.

[34] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

Jing Liang has built several FMCG and young fashion brands in succession. She has created the brand "Squirrel Ai" and the original "360D" branding theory to expand the international influence and reputation of the brand. As a company spokesperson, she has been interviewed by more than a dozen first-tier media, such as CCTV, Hunan TV, Zhejiang TV, and Jiangsu TV, and global media, such as Fortune, Forbes, Financial Times, and MIT TR. She has also created 100 million plays and three million likes on social media platforms, such as Tik Tok. She has been invited to speak at International Joint Conference on Artificial Intelligence (IJCAI), Association for Computer Machinery's special interest group on Knowledge Discovery and Datamining (ACM KDD), Slush, TechCrunch, Union Bank of Switzerland (UBS), and other international summits, and interviewed by Bloomberg and other media.

Ms. Liang is one of the 2019 Top 30 Female Tech Entrepreneurs in China and the 2020 Female Business Leaders of the Year.

Ying Tang (Senior Member, IEEE) received the B.S. and M.S. degrees from the Northeastern University, Shenyang, China, in 1996 and 1998, respectively, and the Ph.D. degree from the New Jersey Institute of Technology, Newark, NJ, USA, in 2001.

She is currently a Full Professor with the Department of Electrical and Computer Engineering, Rowan University, Glassboro, NJ, USA, and the Director of the Institute of Smart Education, Qingdao Academy of Intelligent Industries, Qingdao, China. Her current research interests lie in the area of discrete-event systems and visualization, including virtual reality/augmented reality, modeling and adaptive control for computer-integrated systems, intelligent serious games, green manufacturing and automation, blockchain, and Petri nets. Her work has resulted in one U.S. patent and over 170 peer-reviewed publications, including 56 journal articles, two edited books, and six book/encyclopedia chapters.

Dr. Tang served as an Associate Editor for IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING from 2009 to 2014. She is an Associate Editor of IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS and an Editorial Board Member of *International Journal of Remanufacturing*. She is also a Guest Editor for the Special Issue on Behavioral Modeling, Learning, and Adaptation in Cyber-Physical Social Intelligence in IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, the Special Issue of Intelligent Energy Solutions to Sustainable Production and Service Automation in IEEE TRANSACTIONS ON ENGINEERING SCIENCE AND AUTOMATION, and the Special Issue of Advances in Green Manufacturing and Optimization in Processes. She is the Founding Chair of the Technical Committee on Intelligent Solutions to Human-Aware Sustainability of IEEE Systems, Man and Cybernetic and the Founding Chair of the Technical Committee on Sustainable Production Automation of the IEEE Robotic and Automation.

Ryan Hare (Member, IEEE) is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Rowan University, Glassboro, NJ, USA.

His current research interests include augmenting student learning with adaptive educational systems, improving student engagement with educational games, and using machine learning methods such as reinforcement learning to create more intelligent adaptive tutoring systems.

Ben Wu (Member, IEEE) received the B.S. degree from Nankai University, Tianjin, China, in 2008, and the Ph.D. degree from the Department of Electrical Engineering, Princeton University, Princeton, NJ, USA, in 2015.

He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Rowan University, Glassboro, NJ, USA. His current research interests include signal processing and communication networks.

Dr. Wu is also a member of Optica (formerly OSA). He received the IEEE Photonics Society Graduate Student Fellowship in 2015.

Fei-Yue Wang (Fellow, IEEE) received the Ph.D. degree in computer and systems engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

He joined The University of Arizona, Tucson, AZ, USA, in 1990, where he became a Professor and the Director of the Robotics and Automation Laboratory and the Program in Advanced Research for Complex Systems. In 1999, he founded the Intelligent Control and Systems Engineering Center, Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China, under the support of the Outstanding Overseas Chinese Talents Program from the State Planning Council and "100 Talent Program" from CAS. In 2002, he joined the Laboratory of Complex Systems and Intelligence Science, CAS, as the Director, where he was the Vice President for Research, Education, and Academic Exchanges with the Institute of Automation from 2006 to 2010. In 2011, he was named the State Specially Appointed Expert and the Director of the State Key Laboratory for Management and Control of Complex Systems, Beijing. His current research interests include methods and applications for parallel systems, social computing, parallel intelligence, and knowledge automation.

Dr. Wang was elected as a fellow of INCOSE, IFAC, ASME, and AAAS. He was a recipient of the National Prize in Natural Sciences of China and the Outstanding Scientist by ACM for his research contributions in intelligent control and social computing in 2007, the IEEE TRANSACTIONS ONINTELLIGENT TRANSPORTATION SYSTEMS (ITS) Outstanding Application and Research Awards in 2009, 2011, and 2015, and the IEEE SMC Norbert Wiener Award in 2014. He has been the general or program chair of more than 30 IEEE, INFORMS, ACM, and ASME conferences. He was the President of the IEEE ITS Society from 2005 to 2007, the Chinese Association for Science and Technology, USA, in 2005, and the American Zhu Kezhen Education Foundation from 2007 to 2008. He was the Vice President of the ACM China Council from 2010 to 2011 and the Chair of IFAC TC on Economic and Social Systems from 2008 to 2011. He is the President-Elect of the IEEE Council on RFID. Since 2008, he has been the Vice President and the Secretary-General of the Chinese Association of Automation. He was the Founding Editor-in-Chief of the *International Journal of Intelligent Control and Systems* from 1995 to 2000 and the *IEEE Intelligent Transportation Systems Magazine* from 2006 to 2007. He was the Editor-in-Chief of the IEEE INTELLIGENT SYSTEMS from 2009 to 2012 and the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS from 2009 to 2016. He is also the Editor-in-Chief of the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS and the Founding Editor-in-Chief of the IEEE/CAA JOURNAL OF AUTOMATICA SINICA and the *Chinese Journal of Command and Control*.