Adaptive Difficulty Adjustment for Task Assignment in Online Learning Environments and Adjustment of Remembering Methods

1st Anis Yazidi
dept. of Computer Science
OsloMet - Oslo Metropolitan University
Oslo, Norway
anis.yazidi@oslomet.no

2nd Morten Goodwin
dept. of Computer Science
University of Agder
Kristiansand, Norway
morten.goodwin@uia.no

3rd Hugo Hammer
dept. of Computer Science
OsloMet - Oslo Metropolitan University
Oslo, Norway
hugo.hammer@oslomet.no

4th Asieh Abolpour Mofrad
dept. of Computer Science
OsloMet - Oslo Metropolitan University
Oslo, Norway
asieh.abolpour-mofrad@oslomet.no

5th Erik Arntzen
Dept. of Behavioral Science
OsloMet - Oslo Metropolitan University
Oslo, Norway
erik.arntzen@equivalence.net

***Abstract*—A task assignment method which we reckon as Balanced Difficulty Task Finder (BDTF) is proposed for learning and remembering techniques in traditional methods such as Adjusting Delayed Matching to Sample (DMTS) and Online learning environments. The aim is to assign the tasks with which to maintain the level of motivation and enjoyment during the learning. The idea is similar to Elo's chess skill rating [14] and TrueSkill [15] for matching game players, where "matched players" should possess similar capabilities and skills in order to maintain the level of motivation and enjoyment in the game. The BDTF draws analogy between choosing an appropriate opponent or appropriate game level and automatically choosing an appropriate level of a learning task. Similar to the online learning environments, the BDTF could be an appropriate design solution to other adaptive learning schemes such as adjusting DMTS and spaced retrieval training that can be used for people with memory problems such as people with dementia.**

***Index Terms*—Online Learning, Intelligent Tutoring System, Adaptive Learning, Game Ranking Systems, Adjusting DMTS, SPL**

I. Introduction

Online learning is becoming a significant driving force in today's educational systems. The lack of faculty members is a common trend in today's universities which makes personalized "one to one" teaching challenging, or practically impossible. Students may struggle to fulfill their full potential because the assigned tasks are generic and not tailored to their specific needs and skill level. Several studies show that personalized learning is the key to increased fulfillment of potential [21]. A possible solution to the latter problem is resorting to the advances in *AI* in order to personalize the teaching process. *AI* could be defined as: "*The automation of activities that we associate with human thinking, activities such as decision-making, problem solving and learning*" (Belleman, 1978) [4].

Some of first early studies that allude to the term Intelligent Tutoring System (ITS) dates back to 1982 [31], where D. Sleeman and J.S Brown pioneered the idea of a system designed to help students reach their full potential in a limited amount of time. A few years later, Bloom et al. [5] published a study demonstrating that individual tutoring is twice as effective as group teaching. Later, online teaching courses such as *Kahn Academy*, digital hand in tools like *Fronter*, and plagiarism controls like *Ephorus (Fronter)* have emerged. True *ITS* also exists with open tools like *Codeacademy* and other e-learning platforms.

An *ITS* is supposed to "provide immediate and customized instruction or feedback to learners" [26]. In this paper, we provide algorithms that aspire to fulfil the latter statement for the purpose of task selection.

ITSs can be applied in some traditional learning methods in behavior analysis such as Titrated delayed matching-to-sample (TDMTS) method, also referred as adjusting DMTS [12], [30][1]. TDMTS has been used to study remembering in a variety of settings, including to study important variables in analyzing short-term memory problems [3].

Spaced Retrieval Training (SRT) [8] is another method of learning and retaining a piece of information by recalling that piece of information over increasingly longer intervals. The underlying problem in SRT is also similar to the online task assignment which is addressed here. The SRT method is especially used for people with dementia [7]. In both cases,

[1]Matching-to sample (MTS) procedures, have been frequently used to study complex human behavior (see for instance, [12], [29]). See [2] for an overview of MTS experiments and several variables that can be manipulated when designing an experiment through MTS procedures. In adjusting DMTS, the length of the delay changes as a function of the participants' responses, which makes it similar to the adaptive task assignment problem.

a task finder method which is based on the adaptive task difficulty would be beneficial.

We present a formal theory by which an *ITS* can select the difficulty of task in a similar manner to selecting an opponent of similar capabilities in *balanced* difficulty game [15]. We reckon the method as Balanced Difficulty Task Finder (BDTF). As suggested by systems such as Elo's chess skill rating [14] and TrueSkill [15] for matching game players, matched players should have similar capabilities and skills in order to maintain the level of motivation and enjoyment. We draw analogy between choosing an appropriate opponent or appropriate game level and automatically choosing an appropriate level of a learning task. In the context of game playing, too strong opponent or too weak opponent is a demotivating factor. Similarly, in the context of ITS, too easy exercises or too difficult is a demotivating factor for a student. Please note that, by way of analogy, we can model the student as the player and the chosen task by the system as the opponent. We provide some theoretical results as well as experimental results that demonstrate the convergence of BDTF to an appropriate target difficulty level.

A. Paper Organization

The remainder of this paper is organized as follows. Section II reviews the state of the art. Section III models task selection as balanced difficulty game by resorting to our devised BDTF. Furthermore, we give some theory as well as experimental results that catalogues the convergence properties of the BDTF.

II. State of Art

In this section, relevant studies and papers are discussed to give the reader an overview over the current state of the art. Although several papers on this topic exist dating back several years, the literature reviewed in this section is limited to content published (preferably) after 2005.

There are several approaches to create an *ITS*. In the most recent papers, we are presented with a mix of different artificial intelligence approaches to solve the problem. Common for most of the papers reviewed is the need for a student model including different properties like learning-rate, previous experience and other variables. An approach for such a model *(from now referred to as the* ***student model****)* is represented in some form in numerous studies [6], [10], [11], [22].

The use of the student model in recent papers suggests that this approach is fairly common in the field of *ITS*. Even though the model itself is fairly common, the implementation varies significantly. As an example of this, Clement et al. [10] resort to a combination of a student model and a cognitive model to create a tutoring model. With this approach, the authors try to eliminate the need for a strongly typed student model. The goal is to adjust the learning tasks to individual students with as little information as possible. The use of a Learnong Automata (LA) algorithm enables the system to find the optimal learning sequence for a specific student subject to some constraints such as requiring certain activities to happen before others. A disadvantage of the latter approach is particularly the assumption that some tasks should be carried out in an order. The authors of [10] assume that after task A1 either A2 or B1 need to follow. If a student moves to B1, they can never move "back" to any A-task. This is in most cases a simplification of the learning process, since students should be able to work on several categories and repeat previous categories. As an example, just because a student has moved on from if-sentences to for-loops, it is not correct to say that they should never practice on if-sentences again.

The authors use *partially observable Markov decision process (POMDPs)* for optimization of task selection, which is inspired by Rafferty et al. [27] who used the students acquisition level to propose activities. This method requires the system to assume all students learn in the same way. It is also stated that this approach can be optimal, but requires sophisticated student and cognitive models. In most cases these methods are based on *knowledge tracing-methods (KTM)* which attempt to estimate student knowledge in a parametric manner. In most cases the lack of data causes this form of modeling to be inaccurate. The paper also suggests the use of *POMDPs* for modeling a population of students, not individuals, and this approach has been proven to be suboptimal in an *ITS* setting [10], [19].

On the other hand, several improved versions of the *KTM* have been proposed in the literature. A Representative example is the *Bayesian knowledge tracing (BKT)* with skill-specific parameters for each student. There are strong indicators that *BKT* models accounting for the student variance is superior to the *BK* model [25], [33]. This partially nuances the criticism reported by [10].

A significant number of studies indicate that intrinsically motivated students perform better. Thus, this requires that a good *ITS* keeps motivating the student troughout the whole learning experience. Lumsden, Linda S et al. [20] investigated the optimal strategy for motivating the student, and found that one of the main keystones for a motivational experience is task mastery. This is backed up by [10] who proposes a solution where the student is presented with tasks that are neither too easy nor too hard, but slightly beyond their current abilities. The paper argues that this concurs with theories of intrinsic motivation.

In this article, we propose a solution where each student starts with a predefined *optimal-difficulty* [10] which is adjusted over time based on the student answers. Some students may be more prone to be motivated with challenging tasks, and therefore the overall learning outcome may be more effective for these students. On the other hand, we might find students struggling with the default or *optimal-difficulty*. In these cases the learning-rate should be decreased, allowing these students to participate at a slower pace.

There are several possible alternatives to design an *ITS*. We have looked at several candidates in this study, including *multi armed bandits* [10], *Bayesian-networks* [22] and *neural-networks* [34], each with its own advantages. As mentioned earlier the *student model* is an important part of this *ITS*. In the latter reviewed papers, the *neural network* and *Bayesian-*

network both relied on comprehensive student models, with a solid core of data in order to be able to draw accurate assumptions and decisions. These systems are shown to be reliable and effective, but comprehensive data models are required in order to achieve optimal operation [10]. With the use of LA it is possible to eliminate the need for prior-knowledge about the students. The LA is efficient, and it requires a weaker link between the student and the cognitive model. Clement et al. [10] propose a LA for seven to eight years old school-children learning to decompose numbers while manipulating money. There is no guarantee that a similar approach is viable for use in the context of programming-exercises.

A limited number of studies describe the use of *ITS* in programming courses. As representative studies, we identified *Java Sensei* [34] (sep 2015) and *ASK-ELLE* [17] (2012). Each of the latter studies use a different *machine learning approach*. *Java Sensei* resorts to a combination of *neural-network* strategies and emotion sensors to register information and to make decisions based on input. *ASK-ELLE ITS* utilizes a domain reasoner using a Haskel Compiler called *Helios*. This compiler was developed to give feedback on wrong syntax. The system requires each student to complete a given task, but helps the student accomplish the tasks by giving hints and examples relevant to found error(s).

Even though a generic solution is presented in [10] relying on Multi-Armed Bandit, actual implementations for such a system in the context of programming have not been well-investigated. In this study we propose a solution utilizing the current knowledge of Multi-Armed Bandit to create an ITS for programming tutoring system.

Before moving to the model and contribution of this paper, we refer to the Stochastic Point Location (SPL) problem which has some similarities to the current work. A considerable amount of literature has been published on SPL since the Oommen work [24] (see for instance [23], [32]). In SPL, the problem is to find a point location in a line through the guidance of an external environment which might give faulty advice. Many scientific and real-life problems can be modeled as the instances of SPL problem, including adaptive task assignment problem. For instance, in [23], some authors of this paper discuss that the point location can represent the difficulty level of a task that a participant can handle, and tries to find that point as fast and accurate as possible. The success chance at such a point can be considered as 50%[2]. In other words the model finds a manageable difficulty level and can be used in TDMTS, SRT and online environment. However, in the current work, the aim is not to find the participant level, but to figure out an appropriate level that is motivating and enjoyable, for example with 70% chance of success. In comparison with [23], where the adjustment technique is symmetric, in the current work the effect of correct and incorrect responses are not the same, i.e. the adjustment is asymmetric.

[2]Since more difficult tasks are not manageable and so the tasks will be handled in a random way.

III. Modeling Task Selection as Balanced Game using Balanced Difficulty Task Finder

In this Section, we present the main contribution in this article reckoned as BDTF which is a theory that aspires to learn the appropriate difficulty of a task rather than exploring different types of tasks as in the case of our previous work [1]. Although both approaches can be combined, we clearly distinguish between them as the second case can be seen as a novel theory for determining the adequate difficulty level of an assignment for the purpose of keeping the learning activity "motivating", while the first is more concerned about "exploring" the different tasks in a similar manner to bandit problem.

Difficulty is a subjective concept, or more precisely, it is more "individual" and personal. We argue that difficulty should be tailored to the ability of the student. In fact, as in video games, or chess, the player is motivated by an appropriate level of challenge or equivalently difficulty. For example, the purpose of Xbox TrueSkill system [15] is to match players that have similar capabilities so that the outcome of the game is unpredictable (optimally equi-chance of winning and losing). Elo tries to find a global ranking among players and TrueSkill is similar to the Elo rating system for matching chess players. We advocate that, in a similar manner to TrueSkill and Elo, a student needs to find an enough challenging assignment that matches his capabilities.

We provide a sound mathematical formulation that emanates from the field of stochastic approximation [18].

If students are confronted with too hard assignments compared to their capabilities, they will tend to abandon the learning task and will feel discouraged. On the other hand, too easy assignments do not produce enough positive learning effect and might create a feeling of "boredom".

A. Related work on Games

A representative study that sheds lights on the relationship between three inter-related concepts: difficulty, motivation and learning is presented in [9], [28] and introduces the so-called Flow Channel. According to [9], [28], when the difficulty exceeds the learner's skill, the learner experience a feeling of anxiety at the thought of his learning skills are insufficient, and as a result gets demotivated. Consequently, the learner tends to abandon the activity after short time. On the other hand, boredom takes place in the other extreme case where the student level is much higher than the assignment's difficulty. In this sense, the student perceives the assignment as a waste of time. The ideal case according to [9], [28] takes place when the aptitude of the learner and the difficulty level are in "state of balance". In this case, we say that the learner achieves a state of Flow. Flow is defined by Chen [9] as: "the feeling of complete and energized focus in an activity, with a high level of enjoyment and fulfillment".

As reported in [13], the notion of difficulty in games does not seem to have attracted much attention in the field of education in general. In this perspective, the proposed BDTF tries to bridge the gap between two seemingly disjoint fields of research, namely, ITSs and game ranking/matching systems.

The most pertinent work to our approach emanates from the realm of computer games and chess where it was remarked that when the level of the game is either too difficult or too easy, the players abandon playing [9], [28]

A lot of literature has been centred on the design of adaptive method to adjust the difficulty of the game so that to match the level of the player [16].

B. Formulating Learning as a Balanced Difficulty Game

Without loss of generality, we suppose that the difficulty of any given task can be characterized by a real number from $[0, 1]$, where 0 denotes the lowest possible difficulty and 1 denotes the highest possible difficulty.

The main intuition behind BDTF is the fact that the chance of a student for succeeding in a given task decreases monotonically as the difficulty level increases. We suppose that a student possesses a characterizing skill-curve that describes the relationship between the difficulty of the task and his chance for succeeding in solving the task. We assume that the tasks are ranked on scale from 0 to 1 by an expert such as teacher where 0 denotes the lowest level of difficulty and 1 denotes the highest level of difficulty.

Let λ be an update parameter that is in the interval $]0, 1[$ and let p denote the difficulty of the task. We suppose that s^* is the optimal success probability that we want a learner (student) to experience. It is up to the designer of the intelligent tutoring system to fix the desired target chance of the succeeding in a task for a student. Thus, our approach will try to adjust in an online fashion the difficulty of the given tasks in a manner that drives the system towards a state of flow [9]. Inspired by Elo system, one can choose $s^* = 0.5$ which basically means that the designer desires that the student finds the tasks challenging enough by fixing the target success probability to 50%.

Please note that this reflects the most uncertain case since the outcome of the task in terms of success or failure is unpredictable. However, fixing s^* value requires more in depth study that takes into account many factors including psychological factors. In this paper, and in all the experiments presented in the rest of the article, we will fix $s^* = 0.7$ which basically reflects the fact that we desire the student to succeed most of the time in solving the given task while failing 30% of the time.

In addition, we suppose that we are operating in a discrete time space and let t be the current time instant. The difficulty of the next assignment at time instant $t + 1$ depends on the difficulty of the solved assignment at time instant t as well as the previous achievement (success or failure).

$$p(t+1) = \begin{cases} \min(1, p(t) + \lambda(1 - s^*)) & : \text{if } x(t) = 1 \\ \max(0, p(t) - \lambda s^*) & : \text{if } x(t) = 0 \end{cases} \quad (1)$$

where $x(t)$ denotes the binary variable that records the result of solving the task given at time instant t. $x(t) = 0$ in case of failure and 1 in case of success.

Equation (1) describes a recursive update of the difficulty of the tasks depending on the performance of the student, $x(t)$. According to Equation (1), the difficulty gets increased upon success and decreased upon failure. We suppose that at time $t = 0$, the BDTF starts by suggesting a task with difficulty $p = 0.5$, i.e, we start with tasks with "medium level". We suppose that for a student i, there is function $f_i(p)$ that describes the success probability given the difficulty of the task. Whenever there is no ambiguity, we drop the index i. As explained previously, the latter function is monotonically decreasing. Please note that $x(t) = 1$ with probability $f(p(t))$ and $x(t) = 0$ with probability $1 - f(p(t))$. We will later provide theoretical results that demonstrate that if there exists a point p^* such that $f(p^*) = s^*$ then the update equation converge to it. Since p is defined over $[0, 1]$ and $f(p)$ is decreasing over $[0, 1]$ and admits values in $[0, 1]$, then for any function f_i such point p^* is unique (if it exists). A simple and sufficient condition for the existence as well as uniqueness of p^* is that $f_i(0) = 1$ and $f_i(1) = 0$. This has an intuitive interpretation: the success probability for the $\min$ difficulty is one and for the $\max$ difficulty is zero. However, in the most general case, $f(0)$ might be different from one and $f(1)$ might be different from zero.

The following theorem catalogues the convergence of our scheme for an arbitrary monotonically decreasing function f such that f is mapping from $[0, 1]$ to $[0, 1]$[3].

Theorem 1. *The stochastic process $p(t)$ converges to one of the three following cases as the learning parameter λ tends to zero:*

Case 1: *If* $\min f(p) \le s^* \le \max f(p)$, *then* $\lim_{t\to\infty} \lim_{\lambda\to 0} p(t) = f^{-1}(s^*) = p^*$.

Case 2: *If* $\max f(p) < s^*$, *then* $\lim_{t\to\infty} \lim_{\lambda\to 0} p(t) = 0$.

Case 3: *If* $\min f(p) > s^*$, *then* $\lim_{t\to\infty} \lim_{\lambda\to 0} p(t) = 1$.

Proof: The proof of this theorem is based on the results of the stochastic approximation theory [18] and is omitted for the sake of brevity this paper.

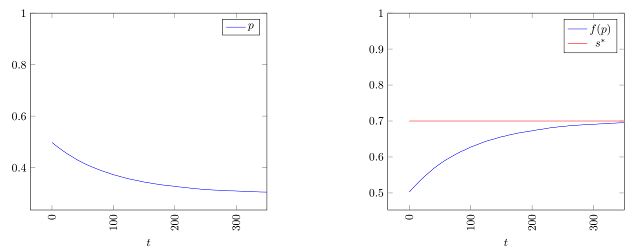
C. Experimental Results

In this section, we provide some experimental results which confirm the theoretical results presented in Theorem 1. In order to describe the relationship between difficulty and success, we define a linear function $f(p) = -a.p + b$, where $a > 0$, ensuring that f is decreasing. Please note that we could have used other types of decreasing functions, such as logistic functions that are usually used in Elo systems. However, the aim of the section is to rather confirm the theoretical properties of our scheme so a simple decreasing function suffices.

Figure 1 depicts the time evolution of p and the corresponding success probability $f(p)$ where $f(p) = -p + 1$ for an update parameter $\lambda = 0.01$. Please note that since $\min f(p) = 0 \le s^* = 0.7 \le \max f(p) = 1$, then according to our Theorem, $p(t)$ converges to $p^* = f^{-1}(s^*) = 0.3$. By

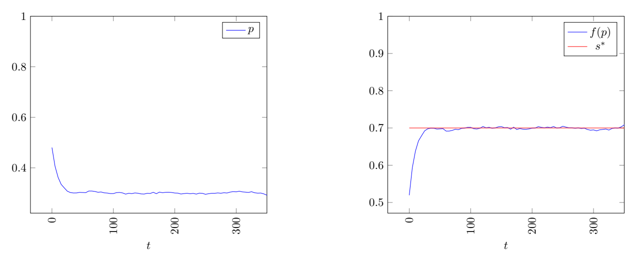
[3]The function $f(.)$ has values within $[0, 1]$ since it denotes the probability of success.

decreasing the update parameter to $\lambda = 0.01$, we remark that the convergence is slower than for the case of Figure 2 where $\lambda = 0.1$.



(a) (b)

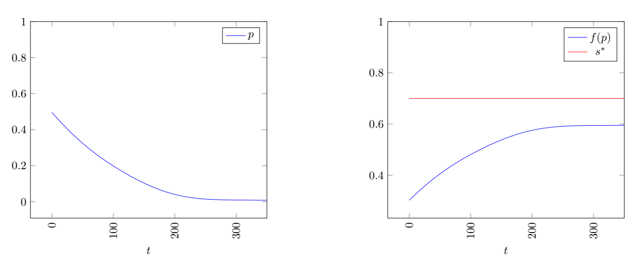
Fig. 1. This figure depicts the time evolution of p and $f(p) = -p + 1$ respectively for $s^* = 0.7$, $\lambda = 0.01$, where (a) Time Evolution of p (b) Time Evolution of $f(p)$.



(a) (b)

Fig. 2. This figure depicts the time evolution of p and $f(p) = -p + 1$ respectively for $s^* = 0.7$, $\lambda = 0.1$, where (a) Time Evolution of p (b) Time Evolution of $f(p)$.

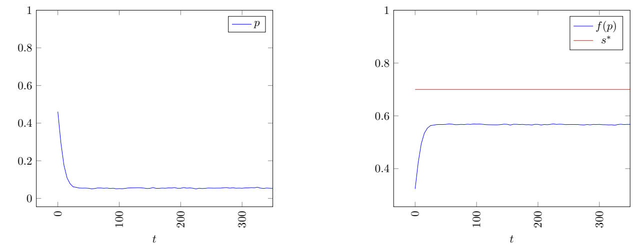
Figure 3 depicts the time evolution of p and the corresponding success probability $f(p)$ where $f(p) = -0.6.p + 0.6$ for an update parameter $\lambda = 0.01$. Please note that since $\max f(p) = 0.6 < s^* = 0.7$, then $p(t)$ converges to $p^* = 0$. The convergence is slower for $\lambda = 0.01$ than the case of Figure 4 where $\lambda = 0.1$. Please note that this represents the second case resulted from the theorem.



(a) (b)

Fig. 3. This figure depicts the time evolution of p and $f(p) = -0.6.p + 0.6$ respectively for $s^* = 0.7$, $\lambda = 0.01$, where (a) Time Evolution of p (b) Time Evolution of $f(p)$.

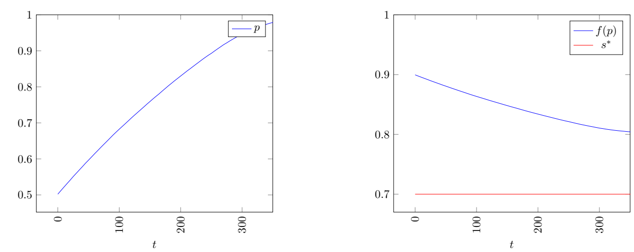
Figure 5 depicts the time evolution of p and the corresponding success probability $f(p)$ where $f(p) = -0.2.p + 1.0$ for an update parameter $\lambda = 0.01$. Please note that since $\min f(p) = 0.8 > s^* = 0.7$, then $p(t)$ converges to $p^* = 1$. The convergence is slower for $\lambda = 0.01$ than the case of Figure



(a) (b)

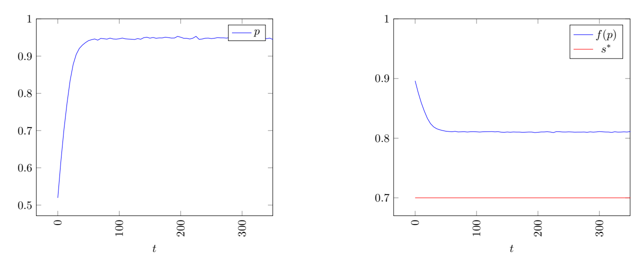
Fig. 4. This figure depicts the time evolution of p and $f(p) = -0.6.p + 0.6$ respectively for $s^* = 0.7$, $\lambda = 0.1$, where (a) Time Evolution of p (b) Time Evolution of $f(p)$.

6 where $\lambda = 0.1$. This represents the third case derived from the theorem.



(a) (b)

Fig. 5. This figure depicts the time evolution of p and $f(p) = -0.2.p + 1.0$ respectively for $s^* = 0.7$, $\lambda = 0.01$, where (a) Time Evolution of p (b) Time Evolution of $f(p)$.



(a) (b)

Fig. 6. This figure depicts the time evolution of p and $f(p) = -0.2.p + 1.0$ respectively for $s^* = 0.7$, $\lambda = 0.1$, where (a) Time Evolution of p (b) Time Evolution of $f(p)$.

As we see in all the figures, the smaller value of λ results a slower, but smoother convergence. Hence, the value of λ can be chosen in a way to prioritize between speed and accuracy at the task assignment.

D. Future work and discussion

The BDTF approach deals only with binary feedback. It is possible to extend our work so that to accommodate non-binary feedback in the form of a continuous or discrete score reflecting the achievement of the student in solving a given task. Furthermore, as a future work, we intend to explore the effect of learning on the progress of the student. Intuitively, the success probability $f(p)$ shall also be frequency dependent, i.e, the more assignments the student tries, the higher the chance of success in future tasks. This is also described as the learning effect that results from repetitive trials. The latter effect can be easily accommodated in our model by rendering $f(p)$ a function of the number of trials, meaning the dynamics of $f(p)$ shall include a frequency dependent term. An interesting avenue for research is the possibility of introducing the recency and spacing in time between the different student trials as an extra parameter in $f(p)$.

IV. Conclusion

In this paper, we tackled the problem of personalized task assignment in online learning environment as well as training methods for retaining information. We present the BDTF which is a formal theory by which an ITS can fine tune the difficulty of a task to a level that matches the student level. The underlying assumption of the BDTF is that the ITSs can fine tune the difficulty of the task to a continuous level. The BDTF application to the learning methods that focus on memory and retaining information such as adjusting DMTS and spaced retrieval training methods is discussed. These methods are looking for the best delay time between two consecutive tasks and can be used for memory training.

References

[1] P.-A. Andersen, C. Kråkevik, M. Goodwin, and A. Yazidi. Adaptive task assignment in online learning environments. In *Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics*, page 5. ACM, 2016.

[2] E. Arntzen. Training and testing parameters in formation of stimulus equivalence: Methodological issues. *European Journal of Behavior Analysis*, 13(1):123–135, 2012.

[3] E. Arntzen and H. S. Steingrimsdottir. On the use of variations in a delayed matching-to-sample procedure in a patient with neurocognitive disorder. In B. Swahn, Palmier, editor, *Mental Disorder*. iConcept Press, 2014.

[4] R. E. Bellman et al. *An introduction to artificial intelligence: Can computers think?* Boyd & Fraser Publishing Company, 1978.

[5] B. S. Bloom. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational researcher*, 13(6):4–16, 1984.

[6] P. Brusilovsky and E. Millán. User models for adaptive hypermedia and adaptive educational systems. In *The adaptive web*, pages 3–53. Springer-Verlag, 2007.

[7] C. J. Camp, J. W. Foss, A. M. O'Hanlon, and A. B. Stevens. Memory interventions for persons with dementia. *Applied Cognitive Psychology*, 10(3):193–210, 1996.

[8] C. J. Camp, G. Gilmore, and P. Whitehouse. Facilitation of new learning in alzheimer's disease. *Memory, aging, and dementia: Theory, assessment, and treatment*, pages 212–225, 1989.

[9] J. Chen. Flow in games (and everything else). *Communications of the ACM*, 50(4):31–34, 2007.

[10] B. Clement, D. Roy, and P.-Y. Oudeyer. Multi-armed bandits for intelligent tutoring systems. *Journal of Educational Data Mining*, 7(2), 2015.

[11] B. Clement, D. Roy, P.-Y. Oudeyer, and M. Lopes. Online optimization of teaching sequences with multi-armed bandits. In *7th International Conference on Educational Data Mining*, 2014.

[12] W. Cumming and R. Berryman. The complex discriminated operant: Studies of matching-to-sample and related problems. *Stimulus generalization*, pages 284–330, 1965.

[13] F. J. Gallego-Durán, R. Molina-Carmona, and F. Llorens-Largo. An approach to measuring the difficulty of learning activities. In *International Conference on Learning and Collaboration Technologies*, pages 417–428. Springer, 2016.

[14] M. E. Glickman. A comprehensive guide to chess ratings. *American Chess Journal*, 3:59–102, 1995.

[15] R. Herbrich, T. Minka, and T. Graepel. Trueskill: A bayesian skill rating system. In *Advances in neural information processing systems*, pages 569–576, 2006.

[16] R. Hunicke. The case for dynamic difficulty adjustment in games. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pages 429–433. ACM, 2005.

[17] J. Jeuring, A. Gerdes, and B. Heeren. Ask-elle: A haskell tutor. In *21st Century Learning for 21st Century Skills*, pages 453–458. Springer, 2012.

[18] H. Kushner and G. G. Yin. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media, 2003.

[19] J. I. Lee and E. Brunskill. The impact on individualizing student models on necessary practice opportunities. *International Educational Data Mining Society*, 2012.

[20] L. S. Lumsden. Student motivation to learn. eric digest, number 92. 1994.

[21] D. Miliband. Personalised learning: building a new relationship with schools. In *Speech by the Minister of State for School Standards to the North of England Education Conference*, 2004.

[22] E. Millán, T. Loboda, and J. L. Pérez-de-la Cruz. Bayesian networks for student model engineering. *Computers & Education*, 55(4):1663–1683, 2010.

[23] A. A. Mofrad, A. Yazidi, and H. L. Hammer. On solving the spl problem using the concept of probability flux. *Applied Intelligence*, pages 1–24, 2019.

[24] B. J. Oommen. Stochastic searching on the line and its applications to parameter learning in nonlinear optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 27(4):733–739, 1997.

[25] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 255–266. Springer, 2010.

[26] J. Psotka, L. D. Massey, and S. A. Mutter. *Intelligent tutoring systems: Lessons learned*. Psychology Press, 1988.

[27] A. N. Rafferty, E. Brunskill, T. L. Griffiths, and P. Shafto. Faster teaching by pomdp planning. In *International Conference on Artificial Intelligence in Education*, pages 280–287. Springer, 2011.

[28] J. Schell. *The Art of Game Design: A book of lenses*. CRC Press, 2014.

[29] M. Sidman. *Equivalence relations and behavior: A research story.* Authors Cooperative, 1994.

[30] M. Sidman. Techniques for describing and measuring behavioral changes in alzheimer's patients. *European Journal of Behavior Analysis*, 14(1):141–149, 2013.

[31] D. Sleeman and J. S. Brown. *Intelligent tutoring systems*. London: Academic Press, 1982.

[32] A. Yazidi, O.-C. Granmo, B. J. Oommen, and M. Goodwin. A novel strategy for solving the stochastic point location problem using a hierarchical searching scheme. *IEEE transactions on cybernetics*, 44(11):2202–2220, 2014.

[33] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized bayesian knowledge tracing models. In *Artificial intelligence in education*, pages 171–180. Springer, 2013.

[34] R. Zatarain Cabada, M. L. Barron Estrada, F. Gonzalez Hernandez, and R. Oramas Bustillos. An affective learning environment for java. In *2015 International Conference on Advanced Learning Technologies (ICALT)*, pages 350–354. IEEE, 2015.