

Uma proposta de sistema de aprendizagem com conteúdo gamificado e com reforço guiado por algoritmos bio-inspirados

Danielli A. Lima¹, Cintia C. Oliveira¹, Lígia C. Pestili¹,
Eduardo C. Silva¹, Maria A. S. Bezerra¹, Henrique A. Lima¹

¹Instituto Federal do Triângulo Mineiro (IFTM) – Campus Avançado Patrocínio
Av. Líria Terezinha Lassi Capuano nº 255 – 38.740-000 – Patrocínio – MG – Brazil

{danielli,cintiaoliveira,ligiapestili,eduardocassiano}@iftm.edu.br,

{masb0705,henriquebrmg}@gmail.com

Resumo. *A aplicação da gamificação no ensino vem ganhando cada vez mais espaço, pois permite a inclusão e cooperação dos estudantes na troca de informações para resolver as atividades dos jogos educativos. Neste trabalho será proposto um ambiente educativo gamificado, onde os conteúdos são apresentados em forma de árvore. Para que se tenha o melhor aproveitamento no aprendizado por reforço, um sistema tutor inteligente irá direcionar o estudante através de um plano de estudos ótimo. Para a confecção desse planejador automático serão utilizados algoritmos baseados em técnicas bio-inspiradas. Resultados mostraram que o sistema tutor inteligente tem boas perspectivas para ser utilizado em cursos, onde a estrutura pode ser modelada através de grafos de hierarquia e o domínio pode ser reforçado com atividades em forma de jogos, garantindo ao aluno um plano de estudos personalizado.*

Abstract. *The gamification application in education is gaining more space because it allows the inclusion and cooperation of students in the exchange of information to resolve the activities of educational games. In this work it will be proposed an environment of education based on gamification, where the contents are presented in a tree. In order to have the best use in learning, an intelligent tutor system will direct the student through a optimal study plan guide. The intelligent tutoring system will be constructed using bio-inspired-based algorithms. Results showed that the intelligent tutoring system has good prospects for use in schools, where the domain structure can be modeled through hierarchy graphs and the domain may be strengthened with games, guaranteeing the student a customized studies plan.*

1. Introdução

Atualmente, há uma grande inserção da tecnologia da informação alinhada ao contexto de ensino e aprendizagem. Para isso, softwares de ensino e aprendizagem são desenvolvidos e embarcados em tablets ou computadores, uma vez que os mesmos podem ser considerados recursos pedagógicos quando proporcionam a complementação dos estudos. Dentro desses estudos, o ramo relacionado à gamificação dos sistemas de ensino tem sido investigado por diversos pesquisadores [Gee 2004], [Kapp 2012] e [Soflano et al. 2015].

O jogo deixou de ser apenas uma forma de entretenimento e tem tomado dimensões maiores, enriquecendo as aulas, tornando-as mais dinâmicas e motivadoras para

os alunos, além de contribuir com o avanço qualitativo no processo de ensino aprendizagem, seja por facilitar o ensino de alguns conteúdos abstratos e de difícil entendimento, ou por reforçar conceitos importantes vistos em sala de aula [Ribeiro 2012]. O processo de reforçar conhecimento pode ser realizado através de sistemas tutores inteligentes [de Souza and de Arruda 2015] dentro do contexto de gamificação.

Sabe-se que o reforço é o elemento mais importante no processo de ensino, mas não é somente a presença de estímulos ou da resposta que leva à aprendizagem, mas a presença das contingências de reforço. O professor é o principal responsável por planejar as contingências de reforço. Segundo [Bruner 1970], o estudante pode ser estimulado na construção do conhecimento através de jogos ou desafios organizados em nível de dificuldade. De acordo com [Vigotski 2003], esse estímulo pode ser dado através do trabalho cooperativo. Assim, a criação de um ambiente de reforço flexível e colaborativo, e ao mesmo tempo individualizado, torna-se importante. Os sistemas tutores inteligentes (STI) baseados em técnicas de Inteligência Artificial são os mais utilizados neste tipo de tarefa. Existem diversas abordagens na modelagem de sistemas tutores inteligentes, dentre elas, podemos citar, redes bayesianas [Gamboa and Fred 2002], redes neurais artificiais [Curilem et al. 2007], ou sistemas baseados em lógica nebulosa [Chrysaftiadi and Virvou 2015].

Baseado neste contexto, este trabalho tem por objetivo (i) a proposição de um protótipo de um ambiente de ensino-aprendizagem, denominado *Doctrina*, onde os conteúdos inseridos no sistema pelo professor devem ser apresentados ao aluno através de jogos e em nível de hierarquia através de um grafo. Além disso, (ii) um sistema tutor inteligente será utilizado para guiar o aprendizado do estudante por meio do plano de reforço automático e será apresentado ao aluno através do agente pedagógico virtual *Dominus*. (iii) Esse plano de atividades será modelado através de algoritmos de computação bioinspirada (CBI), através da otimização por colônia de formigas (ACO) e de um algoritmo genético (AG) multi-objetivo, garantindo ao aluno um plano de estudos personalizado com atividades gamificadas. Por fim, (iv) os principais resultados experimentais serão apresentados para o refinamento deste sistema de reforço automático.

2. Fundamentação teórica

Os STIs podem ser definidos como programas de software que dão suporte às atividades da aprendizagem [Gamboa and Fred 2002]. O principal objetivo dos STIs é proporcionar um ensino adaptado a cada aluno, tentando se aproximar ao comportamento de um professor humano na sala de aula. Podemos observar na Figura 1 uma arquitetura tradicional composta basicamente por quatro componentes que serão detalhados em seguida. Essa arquitetura será a mesma utilizada para o sistema *Doctrina* proposto neste trabalho.

O **modelo do aluno** pode ser definido como um módulo onde as características individuais do aluno estão armazenadas ou modeladas. Segundo [Gavidia and Andrade 2003], um modelo realista do aluno implica numa atualização dinâmica à medida que o sistema avalia o desempenho do estudante. Diversos trabalhos focaram na elaboração e refinamento de um modelo de estudante apropriado, dentre eles podemos citar, [Murray 1998], [Jaques and Vicari 2007], [Mathews et al. 2012] e [Greer and McCalla 2013]. O **modelo do sistema tutor** possui o conhecimento sobre as estratégias e táticas para selecioná-las em função das características do aluno [Self 1988] e apresentá-las ao aluno através de um agente pedagógico. Em STIs um modelo bastante utilizado é o de aprendizado indireto realizado por *coaching* que

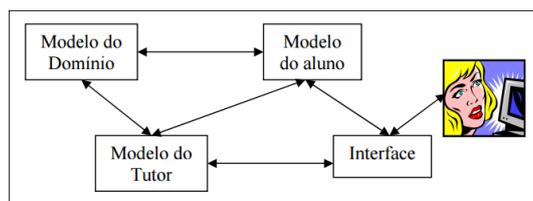


Figura 1. Arquitetura clássica de um Sistema Tutor Inteligente (STI) [Gavidia and Andrade 2003].

aplica atividades como jogos para transmitir conhecimentos relacionados ao conteúdo [Vesselinov and Grego 2012] (gamificação). De acordo com [Huizinga 1967], um jogo é uma atividade voluntária exercida dentro de determinados limites de tempo e espaço, segundo regras livremente consentidas, mas absolutamente obrigatórias, dotado de um fim em si mesmo. Assim, o conceito de **gamificação** tem sido largamente explorado em ambientes educacionais por permitir que o estudante se aproxime semanalmente das atividades o estimulando a atingir um resultado quantificável. A utilização de hipertextos também é muito utilizada em STIs, principalmente para navegação de conteúdos e pode ser utilizada como forma de participação colaborativa (fóruns de discussão) e dinâmica [Wang et al. 2012]. O **modelo do domínio** detém o conhecimento sobre a matéria no formato de regra de produção [McTaggart 2001]. O **modelo da interface** é o responsável pela realização do intermédio da interação entre o tutor e o aluno. Sabe-se que a interface é de fundamental importância para o sucesso de qualquer sistema interativo, e os STIs não são exceção neste requisito [Polson and Richardson 2013].

Um **algoritmo genético** (AG) é uma técnica de busca com o propósito de encontrar soluções aproximadas em problemas de otimização, proposto em [Goldberg and Holland 1988]. AGs são uma classe particular de algoritmos evolutivos que usam técnicas inspiradas pela biologia evolutiva como hereditariedade, mutação, seleção natural e *crossover*. Uma variação para AGs é a utilização de algoritmos genéticos multi-objetivo (AGM) [Deb 1999]. Essa técnica é adequada quando a função de aptidão engloba mais de uma variável para a otimização. No cenário de sistemas tutores inteligentes, vários trabalhos baseados em AGs já foram utilizados [Huang et al. 2007], [Minaei Bidgoli and Punch 2003]. Portanto, a utilização da abordagem de AGs para este trabalho torna-se interessante, principalmente porque a disponibilização do conteúdo para o estudante será dada através de um grafo de hierarquias e o reforço será guiado pela seleção dos conteúdos mais apropriados.

O algoritmo da **otimização por colônia de formigas** (ACO), introduzido por [Dorigo et al. 2008], é uma heurística baseada em probabilidade, criada para solução de problemas computacionais que envolvem procura de caminhos em grafos. As formigas saem de forma aleatória até que, encontrada comida, elas retornam à colônia deixando um rastro de feromônio (feedback positivo). Se outras formigas encontram um desses rastros, elas tendem a seguir a trilha encontrada, retornando e inclusive enfatizando se acharem alimento. Com o transcorrer do tempo, as trilhas de feromônio começam a evaporar, reduzindo, assim, sua força atrativa. Se a situação é de perigo, as mesmas deixam um feedback negativo no ambiente [Semet et al. 2003]. Isso significa, que as formigas vão repelir o feromônio encontrado e vão tentar escolher os caminhos menos visitados. A abordagem utilizando sistemas baseados em colônia de formigas já foi empregada por

diversos pesquisadores em sistemas educativos, dentre esses trabalhos podemos citar, [Graf and Bekele 2006], [Mustafa et al. 2015].

3. Modelo proposto

Nesta seção estão apresentados os principais módulos do sistema Doctrina proposto neste trabalho: (i) modelagem do estudante, (ii) modelagem do sistema de domínio com apresentação de conteúdo em forma de jogos, (iii) modelagem do sistema tutor inteligente, (iv) apresentação do agente pedagógico Dominus. Além disso, para cada um desses módulos será apresentado um protótipo de um **modelo de interface** que representa o meio de comunicação com o estudante.

No sistema Doctrina proposto neste trabalho a **modelagem do estudante** baseia-se no fato de que cada aluno terá um perfil com as características associadas a ele. Neste perfil, estão cadastrados todos os conteúdos cursados e a rotina de estudos adotada. A rotina pode ser classificada em níveis N : lento, normal, rápido $N = \{1, 3, 5\}$ módulos, respectivamente. Além disso, a taxa de aprendizagem T_{ap} de cada aluno será armazenada no perfil do estudante, que representa a quantidade módulos aprendidos e os exercícios realizados. Um exemplo do modelo de estudante está apresentado na Figura 2 à esquerda. Neste modelo um estudante está cadastrado em dois cursos diferentes e possui meta semanal com nível rápido de aprendizado $N = 5$. Isso significa que semanalmente, o estudante deve reforçar o conhecimento através de ao menos cinco lições. O estudante exemplificado na Figura 2 possui uma taxa de aprendizado de 10 unidades. No entanto, se o estudante não fixar o conhecimento com exercícios semanalmente, esta taxa é diminuída.

O **modelo do domínio** detém o conhecimento sobre a matéria no formato de regra de produção. Neste trabalho, o sistema de domínio será adequado para conteúdos onde é possível a aprendizagem por meio de jogos com uma representação por grafos em nível de dificuldade ou hierarquia. Este grafo é do tipo $G = (V, E)$, onde V representa o conjunto de módulos de um determinado curso, o nível de reforço deste módulo ($\rho = \delta + \delta' - \beta$) e o tempo (t) que um determinado módulo já foi realizado, e E representa o conjunto de hierarquias existentes entre cada um desses módulos. O curso deve ser dividido em módulos ou capítulos. Um módulo de conteúdo deve estar interligado à outro módulo através de pré-requisitos. O conteúdo ministrado deve considerar estratégias de aprendizado por meio de reforço. Enquanto um módulo não for realizado pela primeira vez, este módulo não poderá ser reforçado. A ordem de execução de cada módulo deve seguir uma hierarquia de requisitos. Enquanto todos os módulos de um nível não tiverem sido executados, o próximo não fica liberado ao estudante. A regra de produção do modelo de domínio é baseada em ACO. Isso significa que se o estudante não fixar o conteúdo semanalmente a partir da meta pré-estabelecida, a taxa de aprendizado T_{ap} é reduzida semanalmente através de uma constante β , e as barras associadas a cada módulo do curso são ligeiramente evaporadas. Toda vez que o estudante reforça ou realiza um módulo, há um reforço na taxa de aprendizado através de uma constante de depósito δ . Similar ao estímulo de feromônio das formigas. Como os módulos estão interligados através de hierarquia (arestas E do grafo G), o módulo visitado recebe um reforço maior e os módulos adjacentes também recebem uma taxa reforço δ' , sendo esta inferior ao do módulo efetivamente visitado. Um módulo somente recebe um reforço δ' se o mesmo já tiver sido realizado pelo estudante. Se um estudante tem muitos erros ao executar as atividades de um determinado módulo, este não apresentará a barra de aprendizado completa. Isso

significa que a taxa de aprendizado é decrementada através da constante β , quando a atividade não é bem executada pelo estudante. Quando o nível de reforço é totalmente esvaziado, então o estudante deverá refazer este módulo para aumentar a taxa de aprendizado. O sistema tutor inteligente é o responsável por escalonar um plano de atividades ótimo para o reforço do estudante. No exemplo da Figura 2 central, tem-se um curso de programação procedimental. Este curso tem oito módulos, cada módulo pode ser reforçado em até quatro níveis. Essa ideia é similar ao feromônio das formigas forrageiras, que depositam feromônio ao visitar uma região. Além disso, o estudante cadastrado em um determinado curso poderá participar de um fórum, que tem como objetivo o compartilhamento de ideias entre estudantes e o professor responsável pelo curso. Cada estudante poderá ser mencionado através do seu nome de usuário único por outros participantes matriculados num mesmo curso, promovendo um sistema de comunicação mais colaborativo entre eles.

Sabe-se que o reforço por si só não consegue fazer com que o aluno aprenda de maneira eficiente, mas um sistema de reforço alinhado a uma boa estratégia de ensino-aprendizagem pode garantir um aprendizado mais adequado para o estudante. Essa estratégia de ensino aprendizagem denominada de **sistema tutor inteligente** se dá por meio de um planejador semanal de atividades. Esse planejador será apresentado por um agente pedagógico virtual chamado Dominus (Figura 2, à direita) que fará a intermediação entre o sistema Doctrina e o estudante através da interface gráfica. Ou seja, o agente pedagógico Dominus é o responsável por estimular o estudante por meio de um plano semanal de atividades automático. Esse plano semanal de atividades, será escalonado a partir das regras definidas no modelo do estudante e domínio. O plano de atividades será escalonado através de um algoritmo genético multi-objetivo e algoritmo de colônia de formigas. O resultado dessa execução é apresentado na Figura 2, à direita. Se o estudante seguir semanalmente o plano de atividades escalonado automaticamente pelo planejador do STI, o mesmo terá uma resposta positiva à aprendizagem T_{ap} . As contingências de reforço também são baseadas no conceito de gamificação, já que motivam o aluno a estabelecer um ritmo de estudos semanalmente através do cumprimento da meta apresentada pelo planejador automático. Além disso, cada atividade de reforço tem o conteúdo disposto na forma de jogos, que são atividades estruturadas, praticadas com fins recreativos e por isso são considerados instrumentos educacionais pertinentes ao contexto deste trabalho.

Os módulos de conteúdo do sistema Doctrina são apresentados ao aluno através de um plano de atividades semanal pelo agente virtual Dominus. O planejamento automatizado representa um processo que escolhe e organiza módulos de conteúdo para o estudante, de forma que o mesmo possa atingir um resultado ótimo. Neste problema, o resultado ótimo é propor um plano de atividades que tenha a menor nível de barras reforço e que simultaneamente esteja há mais tempo sem reforço pelo estudante. Ou seja, quanto mais feromônio tem um módulo, menor é a chance do mesmo ser escalonado para reforço. Sabe-se que problemas de otimização em grafos são do tipo NP-completos. Isso significa que não existe um algoritmo de tempo polinomial que realize essa tarefa, à medida que os nós do grafo (módulos) aumentam. Para resolver este problema, neste trabalho será utilizado um AG, de tempo polinomial, que gera um plano de atividades automático tentando aproximar-se do ótimo. Esse AG repele as taxas de feromônio mais altas e os módulos (nós do grafo) mais recentemente visitados ($\rho = \delta + \delta' - \beta, t$). Para a otimização, a função-objetivo $f_{obj} = \min(\rho, t)$ tem como meta encontrar os N nós mínimos, da meta do modelo de estudante, e apresentar através de um plano de atividades otimizado. Neste

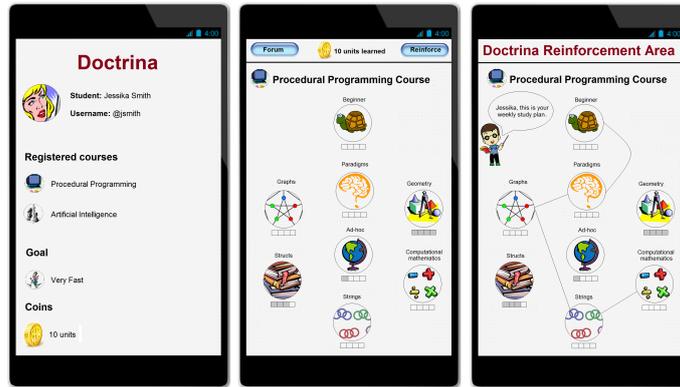


Figura 2. Interface do sistema Doctrina para dispositivo móvel: à esquerda é apresentado o esquema para o modelo de estudante, ao centro são apresentados os conteúdos de um curso de programação, à direita são apresentados o agente virtual Dominus e o resultado da execução do algoritmo genético multi-objetivo para a obtenção do plano com atividades gamificadas de reforço.

caso, o indivíduo é uma representação do espaço de busca do problema a ser resolvido, ou seja, o indivíduo tem exatamente o tamanho N da meta semanal do estudante. Dentro desse cromossomo tem-se dois valores, um que representa a quantidade de feromônio ρ (nível das barras de cada módulo) e o tempo que cada módulo foi executado t . Nenhum cromossomo pode ter valores idênticos, pois deseja-se encontrar N módulos distintos para ser reforçado. A seleção F_{sel} é realizada por meio de diferentes funções onde os indivíduos são ordenados de acordo com a função objetivo e lhes são atribuídas probabilidades decrescentes de serem escolhidos. A escolha é feita através da heurística provida pela função objetivo f_{obj} . A reprodução é realizada através da função de *crossover* F_{cro} de dois indivíduos e da mutação.

4. Resultados

Nesta seção serão apresentados os resultados da execução de configuração do algoritmo genético para a geração do escalonador do plano de atividades do aluno, que tem tamanho N . O AG busca uma solução ótima a partir das constantes de feromônio ρ e t que fazem parte do modelo de domínio, que devem consideradas para que o aluno tenha o plano de atividades ótimo. Inicialmente é apresentado um exemplo onde todos os módulos M de um determinado curso C para um aluno S matriculado. Na Figura 3, temos um exemplo de uma execução do AG, onde os pontos em azul representam os todos os módulos $M = 10$ de um curso C já cumpridos por um estudante. Esse estudante possui um ritmo de aprendizado semanal de $N = 5$. Em vermelho temos os pontos relativos a uma execução intermediária do AG (não-ótima) e em verde temos a execução final do AG (ótima).

Em seguida um experimento foi elaborado para a configuração das constantes do AG a partir dos dados da Figura 3, dentre elas, tamanho da população T_{pop} , a taxa de *crossover* T_{cro} e a taxa de mutação T_{mut} . Cada variação foi executada por 100 vezes e cada execução tem 10 gerações. O tamanho da população foi variado em 7 valores distintos $T_{pop} = \{5, 10, 20, 40, 60, 80, 100\}$, com $T_{cro} = 60$ e $T_{mut} = 5$ fixos. A taxa de *crossover* foi variada em $T_{cro} = \{5, 10, 20, 40, 60, 80, 100\}$, com $T_{pop} = 20$ e $T_{mut} = 5$ fixos. A taxa de mutação foi variada em $T_{mut} = \{0, 5, 10, 20, 30, 40, 50\}$, com $T_{pop} = 20$ e $T_{cro} = 60$ fixos. Para efeito de visualização, os dados gerados no experimento foram plotados no

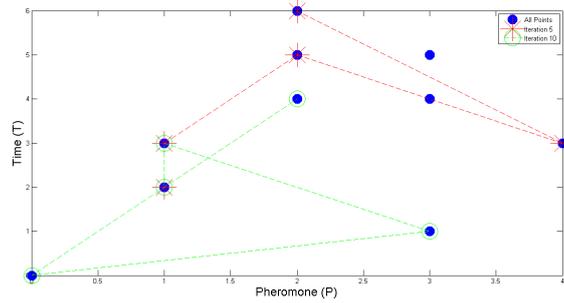


Figura 3. Pontos em azul $M=(p,t)$ de um curso C com $N=5$. Em vermelho tem-se uma execução intermediária do AG e em verde tem-se a execução final do AG (ótima).

gráfico da Figura 4. O gráfico representa a porcentagem de execução de respostas corretas (eixo x) e o valor absoluto de valores ótimo encontrados (eixo y). No eixo z temos os valores referentes à cada variação T_{pop} (azul), T_{cro} (vermelho) e T_{mut} (verde). Através do gráfico é possível perceber que para as taxas T_{pop} e T_{cro} à medida que as mesmas aumentam, elas geram uma melhoria para que o AG encontre soluções ótimas. No entanto, quanto maiores são os tamanhos de população e de taxa de *crossover* mais lenta torna-se a execução do AG. Quando a taxa de mutação T_{mut} é aumentada gradativamente, o valor de acerto começa a se degradar, tornando a busca mais aleatória, o que pode prejudicar o AG encontrar as soluções ótimas dentro do espaço de busca.

Por fim, o último experimento apresenta a variação dos métodos de construção

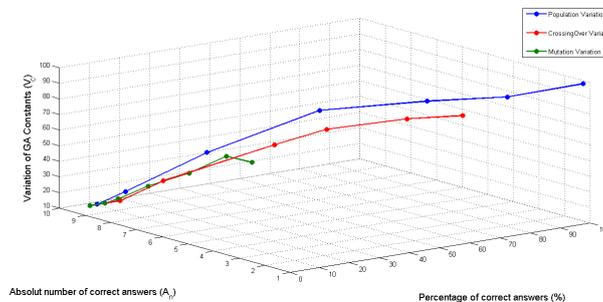


Figura 4. Gráfico do experimento da variação do tamanho da população, da taxa de *crossover* e a taxa de mutação.

das principais funções utilizadas neste trabalho. Cada variação foi executada por 100 vezes e cada execução tem 10 gerações, com $T_{pop} = 20$, $T_{cro} = 60$ e $T_{mut} = 5$. O experimento está apresentado no gráfico da Figura 5, tal que a linha azul representa a variação da função de seleção F_{sel} de pais utilizando quatro abordagens: método de roleta, torneio elitista de tamanho 2, torneio elitista de tamanho 3 e a roleta com ranking linear [Sakawa 2002], que obtiveram, respectivamente, 65, 56, 57, 60. Isso significa que o método de roleta obteve o melhor resultado dentre as funções de seleção implementadas neste trabalho. A linha em azul representa a variação da função de *crossover* F_{cro} utilizando duas abordagens: *crossover* cíclico e *crossover* PMX [Starkweather et al. 1991]. A partir da melhor configuração de acerto da F_{sel} (método de roleta com *ranking* linear), foi iniciada a alteração de F_{cro} . Para este problema, a função de *crossover* cíclico apre-

sentou 55 instâncias de soluções ótimas, enquanto que o *crossover* PMX obteve apenas 65 soluções ótimas. Isso significa que para este problema, o método de *crossover* mais adequado é o PMX aplicado juntamente ao método de seleção por roleta.

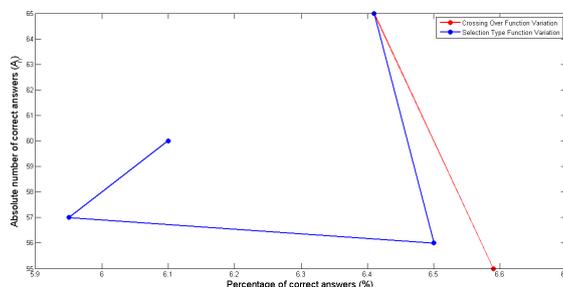


Figura 5. Gráfico do experimento da variação da função de *crossover* e a da função de seleção.

5. Conclusões e trabalhos futuros

Neste trabalho, um sistema tutor inteligente denominado Doctrina foi proposto como forma de ensino-aprendizagem onde o conteúdo pode ser preferencialmente aprendido por meio de reforço e atividades gamificadas. Este sistema possui 4 módulos principais: modelo do estudante, modelo de domínio, modelo de sistema tutor e o modelo de interface, similar ao sistema básico apresentado em [Gavidia and Andrade 2003]. O modelo do estudante contempla as características do estudante, tais como, a taxa de aprendizagem, cursos matriculados, e a meta semanal deste estudante. Este módulo é o responsável por representar o conhecimento e as habilidades cognitivas do aluno em um dado momento. Neste trabalho, o modelo do aluno é realista, pois implica numa atualização dinâmica à medida que o sistema avalia o desempenho do aluno (taxa de aprendizagem). O conteúdo deve ser apresentado em uma sequência hierárquica de conhecimento com requisitos e pré-requisitos (grafo). Cada nó do grafo representa um módulo de conteúdo e este módulo possui valores de reforço associados. Esses valores de reforço foram modelados através do depósito de feromônio, similar à tarefa de forrageamento das formigas. Se o estudante deixa de praticar as atividades, o grau do reforço de cada módulo é evaporado.

O STI é o responsável por prover ao estudante um plano de atividades que seja reforçado semanalmente. Este sistema escalona de maneira ótima um plano de atividades que tem como objetivo escolher os módulos com as menores taxas de feromônio ACO e que não tenham sido recentemente visitados. Para a execução desta tarefa, um AG que executa esta tarefa em tempo polinomial baseado em heurísticas associadas ao problema foi utilizado. Assim que o plano de atividades ótimo é escalonado através do AG, um agente virtual pedagógico Dominus apresenta o plano semanal ao estudante. A partir dos resultados de refinamento deste AG, foi possível encontrar uma configuração adequada à solução deste problema. Adicionalmente, um protótipo de um modelo de interface para dispositivos móveis foi apresentado. Através dessa interface é possível estabelecer um meio de comunicação, como um ambiente de solução de problemas que dá suporte ao aluno nas tarefas disponíveis. Além disso, um protótipo de um módulo de sistema colaborativo, através de um fórum de discussão, foi apresentado como um recurso para auxiliar o estudante durante o desenvolvimento das atividades.

A grande vantagem de se usar o modelo Doutrina proposto, no contexto de ensino-aprendizagem, encontra-se no fato que o modelo é flexível à entrada de parâmetros e o mesmo pode agregar facilmente outros parâmetros inseridos no sistema. Entretanto, não podemos garantir que apenas os comportamentos inseridos por si só já comprovam com exatidão uma melhoria da taxa de aprendizagem do estudante. Mas, elas iniciam uma análise que pode ser aprofundada em um trabalho de pesquisa posterior.

Referências

- Bruner, J. S. (1970). Uma nova teoria de aprendizagem. *Rio de Janeiro: Bloch*.
- Chrysafiadi, K. and Virvou, M. (2015). Fuzzy logic for adaptive instruction in an e-learning environment for computer programming. *Fuzzy Systems, IEEE Transactions on*, 23(1):164–177.
- Curilem, S. G., Barbosa, A. R., and de Azevedo, F. M. (2007). Intelligent tutoring systems: Formalization as automata and interface design using neural networks. *Computers & Education*, 49(3):545–561.
- de Souza, I. D. and de Arruda, B. C. (2015). Gamificação: o aprendizado de idiomas com aplicativos para dispositivos móveis. *Revista Interdisciplinar de Tecnologias na Educação*, 1:191–200.
- Deb, K. (1999). Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary computation*, 7(3):205–230.
- Dorigo, M., Birattari, M., Blum, C., Clerc, M., Stützle, T., and Winfield, A. (2008). *Ant Colony Optimization and Swarm Intelligence: 6th International Conference, ANTS 2008, Brussels, Belgium, September 22-24, 2008, Proceedings*, volume 5217. Springer.
- Gamboa, H. and Fred, A. (2002). Designing intelligent tutoring systems: a bayesian approach. *Enterprise Information Systems III. Edited by J. Filipe, B. Sharp, and P. Miranda. Springer Verlag: New York*, pages 146–152.
- Gavidia, J. J. Z. and Andrade, L. C. V. d. (2003). Sistemas tutores inteligentes. *Rio de Janeiro: COPPE*.
- Gee, J. P. (2004). Learning by design: Games as learning machines. *Digital Education Review*, (8):15–23.
- Goldberg, D. E. and Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99.
- Graf, S. and Bekele, R. (2006). Forming heterogeneous groups for intelligent collaborative learning systems with ant colony optimization. In *Intelligent Tutoring Systems*, pages 217–226. Springer.
- Greer, J. E. and McCalla, G. I. (2013). *Student modelling: the key to individualized knowledge-based instruction*, volume 125. Springer Science and Business Media.
- Huang, M.-J., Huang, H.-S., and Chen, M.-Y. (2007). Constructing a personalized e-learning system based on genetic algorithm and case-based reasoning approach. *Expert Systems with Applications*, 33(3):551–564.

- Huizinga, J. (1967). *Homo ludens: A study of the play-element in culture*. Beacon Press.
- Jaques, P. A. and Vicari, R. M. (2007). A bdi approach to infer students emotions in an intelligent learning environment. *Computers and Education*, 49(2):360–384.
- Kapp, K. M. (2012). *The gamification of learning and instruction: game-based methods and strategies for training and education*. John Wiley & Sons.
- Mathews, M., Mitrovic, A., Lin, B., Holland, J., and Churcher, N. (2012). Do your eyes give it away using eye tracking data to understand students attitudes towards open student model representations. In *Intelligent tutoring systems*, pages 422–427. Springer.
- McTaggart, J. (2001). Intelligent tutoring systems and education for the future. *512X Literature Review April*, 30(2).
- Minaei Bidgoli, B. and Punch, W. F. (2003). Using genetic algorithms for data mining optimization in an educational web-based system. In *Genetic and Evolutionary Computation GECCO 2003*, pages 2252–2263. Springer.
- Murray, T. (1998). Authoring knowledge-based tutors: Tools for content, instructional strategy, student model, and interface design. *The journal of the learning sciences*, 7(1):5–64.
- Mustafa, H. M., Al-Hamadi, A., Abdulrahman, M., Mahmoud, S., and Sarhan, M. O. (2015). On comparative analogy between ant colony systems and neural networks considering behavioral learning performance. *Journal of Computer Sciences and Applications*, 3(3):79–89.
- Polson, M. C. and Richardson, J. J. (2013). *Foundations of intelligent tutoring systems*. Psychology Press.
- Ribeiro, A. L. (2012). Aquisição da escrita na era virtual: incorporando os jogos digitais online. *Domínios de Linguagem*, 6(2):111–127.
- Sakawa, M. (2002). Foundations of genetic algorithms. In *Genetic Algorithms and Fuzzy Multiobjective Optimization*, pages 11–27. Springer.
- Self, J. (1988). *Artificial intelligence and human learning*. Chapman and Hall.
- Semet, Y., Lutton, E., and Collet, P. (2003). Ant colony optimisation for e-learning: Observing the emergence of pedagogic suggestions. In *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*, pages 46–52. IEEE.
- Soflano, M., Connolly, T. M., and Hainey, T. (2015). An application of adaptive games-based learning based on learning style to teach sql. *Computers & Education*, 86:192–211.
- Starkweather, T., McDaniel, S., Mathias, K. E., Whitley, L. D., and Whitley, C. (1991). A comparison of genetic sequencing operators. In *ICGA*, pages 69–76.
- Vesselinov, R. and Grego, J. (2012). Duolingo effectiveness study. *City University of New York, USA*.
- Vigotski, L. S. (2003). *A formação social da mente: o desenvolvimento dos processos psicológicos superiores*. Martins Fontes.
- Wang, S., Fischer, P. M., Jonassen, D. H., and Mandl, H. (2012). *Designing hypermedia for learning*, volume 67. Springer Science and Business Media.