Geração automática de avaliações utilizando o algoritmo ABC para definição da próxima questão

Alexandre Ibrahim Direne[1], Andrey Ricardo Pimentel[1], Eduardo Jaques Spinosa[1], Leonardo Emanuel Melniski[1], Rita de Cassia Tesseroli[1]

[1]Departamento de Informática
Universidade Federal do Paraná

{alexd, andrey, spinosa, lemelniski, rctesseroli}@inf.ufpr.br

Abstract. *The article focuses on the automatic ordinal classification of problem statements of exams in educational contexts. Past research is presented in a critical way, showing that there have been no approaches to statement ordering based on meta-heuristics. In this research, the order of statements shown to a learner is computed by the heuristic algorithm ABC (Artificial Bee Colony) for generating personalized evaluations based on teacher-defined requirements that are adaptive in relation to the domain setting. An empirical study has been carried out by means of executing the ABC guided by two different heuristics of fitness over data on problem statements. The results are portrayed and discussed in a comparative format.*

Resumo. *O artigo enfoca a classificação ordinal automática de questões de provas em contextos educacionais. Diversas pesquisas do passado são apresentadas no texto, mostrando que nenhuma delas abordou a decisão de ordem de cada questão com base em algoritmos fundamentados em mataheurísticas. No presente artigo, a ordem de cada questão apresentada ao aluno é dada pelo algoritmo heurístico ABC (Artificial Bee Colony) de geração de avaliações personalizadas com base em requisitos definidos pelo professor e adaptável ao cenário de um domínio estipulado. Um estudo empírico foi conduzido por meio da execução do ABC guiado por duas heurísticas de aptidão diferentes. Os resultados são apresentados e discutidos de forma comparativa.*

1. Introdução

O processo de ensino-aprendizagem possui como um dos elementos reguladores da aprendizagem a avaliação de estudantes através de provas orais e escritas, relatórios, entre outros [Hoffmann 2001]. A utilização de computadores como instrumento investigativo do aprendizado pode ser feita através da geração automática de avaliações e métricas de mensuração do desempenho do aluno. Dentre os métodos utilizados para a escolha do sequenciamento de questões da avaliação cita-se a teoria de autômatos [Morais 2002, Zanella 2005], redes bayseanas [Almond et al. 2007] e cadeias de markov [Desmarais and Baker 2011]. Na literatura não há pesquisas que propõem o uso de meta-heurísticas inspiradas na inteligência coletiva [1] para a escolha da próxima questão na

[1]Na área da computação o termo inteligência coletiva ou *swarm intelligence* diz respeito ao uso de algoritmos ou dispositivos distribuídos inspirados no comportamento coletivo de insetos sociais e outras sociedades animais na solução de problemas [Augustus 2009].

DOI: 10.5753/cbie.wcbie.2016.407

definição de uma sequência de questões da avaliação. Acredita-se que tal oportunidade deve-se ao fato que as pesquisas atuais trabalham apenas com banco de questões e conhecimentos avaliados pequenos.

Do ponto de vista pedagógico o objetivo da avaliação é investigar a trajetória de construção do conhecimento do estudante [Hoffmann 2001]. Avalia-se para saber se o aluno está ou não aprendendo e qual a distância que o estudante está do objetivo proposto na ementa. Em seguida, utilizando o diagnóstico encontrando é possível criar novas estratégias pedagógicas visando o desenvolvimento do estudante [Demo 2002]. Nesta visão é importantíssimo que as ferramentas avaliativas investiguem o conhecimento do estudante de forma adequada, valorizando o erro do aluno ao invés de frustrá-lo. Isso pode ser feito estimulando-o à resolução das questões e não à memorização de respostas. Para que isso aconteça a avaliação deve encorajar o ato de pensar e possuir uma sequência adequada de questões que favoreça a investigação de conceitos.

Por exemplo, dado o cenário escolar da aplicação da provinha brasil no segundo ano do ensino fundamental [2], supõe-se que a instituição de ensino utiliza uma ferramenta computacional para realização da avaliação preparatória da provinha brasil. Essa ferramenta contém o banco de questões de todos os anos que a provinha brasil foi aplicada e os conteúdos que cada questão avalia. A criação do material preparatório da avaliação por parte do professor é uma tarefa cotidiana, mas analisando computacionalmente é necessária a representação do que será avaliado e como será avaliado. Esses requisitos são desafiadores se analisarmos a necessidade da utilização de um modelo de domínio bem definido e um banco de questões extenso e a partir desses dados identificar uma heurística para geração do sequenciamento de questões.

O presente trabalho utiliza-se da estrutura proposta por [Tesseroli 2015] para modelar o conhecimento e propõe uma heurística para geração automática de avaliações baseada na meta-heurística de colônia de abelhas, *Algoritmo Artificial Bee Colony*.

2. Resenha literária

As ferramentas computacionais que visam resolver o problema da geração automática de questões adaptativas utilizam-se de teorias cognitivas e computacionais para criação de heurísticas de escolha da próxima questão. [Leão and Reis 2006] propõem o gerenciamento e edição gráfica de questões através da ferramenta AvaWeb. Foi desenvolvido um sistema de autoria de avaliações, que dispõe de um ambiente para criar, executar e apresentar a avaliação para o aluno. Utiliza-se a gramática de grafos, onde a conexão entre as questões é através de arestas definidas como resposta certa e errada, é possível determinar qual a próxima questão.

[Zanella 2005] apresenta a geração automática de avaliação pela ferramenta NetAula. É possível a criação de provas, exercícios ou testes utilizando questões cadastradas em um banco de questões e catalogadas quanto à dificuldade, tipo e módulo de aula. A geração da avaliação pode ser automática onde o professor define a quantidade de questões, módulo de aula, dificuldade da questão e tipo.

As ferramentas encontradas pela presente pesquisa permitem a criação de avaliações através de sistemas de autoria. Preocupam-se em caracterizar os possíveis

[2] http://provinhabrasil.inep.gov.br/

dados das questões e criar automaticamente o sequenciamento das questões, de acordo determinados parâmetros. Identifica-se nos trabalhos citados uma lacuna não explorada: a geração automática de avaliações em situações onde a base de questões e de conhecimento é vasta. Os algoritmos de geração automática de avaliações não consideram o tempo de processamento e o tamanho da base de questões e conceitos.

Visto a oportunidade na criação de uma heurística para geração de avaliações o presente trabalho utiliza-se de trabalhos realizados anteriormente [Tesseroli 2015] para definir a estrutura do conhecimento e novas pesquisas para definição da heurística de definição da próxima questão.

3. Conceitos utilizados

3.1. Hierarquia de conceitos

Gagné propõe que as habilidades aprendidas pelo indivíduo são ordenadas e relacionadas através da dependência de aprendizado entre elas. Uma habilidade intelectual geral é composta por um conjunto de habilidades mais simples. As habilidades mais simples são pré-requisitos para aprendizagem da habilidade intelectual geral. A Figura 1 apresenta o mapa de hierarquia de aprendizagem de Gagné. As unidades básicas e específicas são aprendidas primeiro para então evoluir para as capacidades subordinadas, intermediárias. Por último, as capacidades mais gerais, de nível superior, serão aprendidas [Driscoll 2005, Pimentel 2006, Taveira 2003, Magalhães and Carpinteiro 2004].

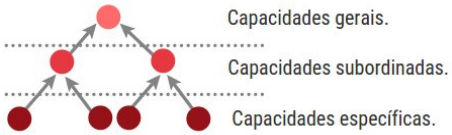

Figura 1. Hierarquia de aprendizagem proposta por Gagné.

3.2. Algoritmo ABC

O algoritmo ABC modela o comportamento natural das abelhas na procura por alimento. Ele é baseado em uma meta-heurística de enxame, mais especificamente a que tenta imitar o comportamento coletivo de abelhas na procura de alimentos [Karaboga and Akay 2009]. É tipicamente utilizado para resolver problemas de otimização. No ABC a posição de uma fonte de alimento representa uma possível solução do problema e a quantidade de néctar de uma fonte de alimento corresponde à qualidade ou medida de aptidão da solução associada [Serapião 2009].

4. Proposta para geração automática de avaliações

Para o desenvolvimento desse trabalho é determinada a estrutura da base de conhecimentos e uma base de questões. A base de conhecimentos é a hierarquia de conceitos de um domínio de conhecimento. A base de questões são as perguntas da avaliação. As questões avaliam conceitos pertencentes a hierarquia de conceitos e possuem um grau de complexidade definido para cada função *fitness*. A Figura 2 apresenta a modelagem básica da proposta. Por exemplo, dada uma base de conhecimentos com a seguinte hierarquia de

conceitos *C1, C2, C3, C4*. As questões *Q1, Q2, Q3*, pertencentes a base de questões, investigam um ou mais conceitos determinados na base de conhecimentos. A heurística seleciona a próxima questão baseada em características de proximidade dos conceitos e complexidade das questões.

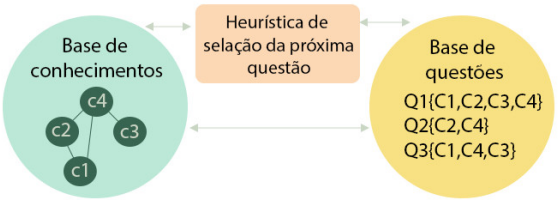

Figura 2. Proposta de modelagem

O presente trabalho optou pela estrutura de conhecimentos proposta por [Tesseroli 2015]. Esta define a base de conhecimento a partir da hierarquia de aprendizagem proposta por Gagné. Cada nó da hierarquia é um conceito, esses nós relacionam-se entre si através de setas direcionadas. Na base da hierarquia estão os conceitos mais específicos e aprendidos inicialmente. Os conceitos específicos estão associados com os conceitos intermediários e gerais, presentes no meio e topo da hierarquia. Uma questão investiga conceitos necessários para sua resolução, estes conceitos estão expressos na forma de nós na hierarquia. A Figura 3 denota o exemplo de uma base de conhecimentos com conceitos de 1 a 10 e a investigação dos conceitos 3, 6, 9, 10 pela questão 01.

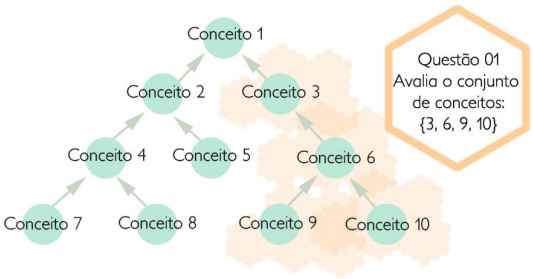

Figura 3. Visão geral da relação entre conceitos e questões

Dada uma questão inicial, com um conjunto de conceitos avaliados e grau de complexidade, a próxima questão da avaliação investiga um conjunto de conceitos semelhantes e possui grau de complexidade com valor próximo. A métrica utilizada para seleção da próxima questão utilizando esses parametros é o algoritmo ABC, disponibilizado pelo pacote *ABCoptim* [3], implementado no R[4].

O algoritmo ABC modela o comportamento natural das abelhas na procura por alimento. Este algoritmo é baseado em uma meta-heurística de enxame, especificamente o comportamento coletivo de abelhas na procura de alimentos [Karaboga and Akay 2009]. No ABC a posição de uma fonte de alimento representa uma possível solução do problema e a quantidade de néctar de uma fonte de alimento corresponde à qualidade ou medida de aptidão da solução associada, equivalente ao valor da função-objetivo [Serapião 2009].

[3] *ABCoptim* – `https://cran.r-project.org/web/packages/ABCoptim/index.html`
[4] *The R Project for Statistical Computing* – `https://www.r-project.org/`

Abaixo são listados os parâmetros para criação da heurística de seleção da próxima questão:

- *A questão resultante da iteração deve ter grau de complexidade próximo a questão inicial:* a próxima questão da avaliação deve ser ligeiramente mais difícil que a anterior.
- *Os conceitos avaliados na questão resultante devem ser similares aos da questão inicial:* a próxima questão da avaliação deve avaliar os conceitos da questão anterior mais um conceito que possui dependência de aprendizado dos anteriores.

Para a criação das funções *fitness* estipulou-se duas formas de cálculo do grau de complexidade das questões. A primeira equação do cálculo do grau de complexidade da questão é a formula *Grau_Complex*$_1$. Esta é a somatória das alturas dos nós com o grau de entrada dos nós.

$$Grau_Complex_1 = \sum_{nó \in C_q} (altura(nó) + grau_entrada(nó)) \tag{1}$$

A segunda equação do cálculo do grau de complexidade da questão é a formula *Grau_Complex*$_2$. Esta é a somatória das alturas do nós com o grau dos nós.

$$Grau_Complex_2 = \sum_{nó \in C_q} (altura(nó) + grau(nó)) \tag{2}$$

A *fitness 1* é a razão entre dois fatores. O fator dividendo é a somatória do cálculo de $Grau_Complex_1$ da diferença entre os conjuntos C_r e C_e. Onde C_r representa o conjunto dos conceitos avaliados na questão resultante, enquanto, C_e representa o conjunto dos conceitos avaliados na questão inicial. O fator divisor é o tamanho da intersecção entre C_r e C_e.

$$fitness_1 = \frac{\sum(Grau_Complex_1(C_r - C_e))}{|C_r \cap C_e|} \tag{3}$$

A *fitness 2* é definida pela razão entre dois fatores. O fator dividendo é a soma de duas equações; a primeira equação é o somatório do cálculo do $Grau_Complex_2$ da diferença do entre os conjuntos C_r e C_e; e a segunda equação é o somatório das menores distâncias entre os vértices do conjuntos C_r e C_e. O fator divisor é o tamanho da intersecção entre C_r e C_e.

$$fitness_2 = \frac{\sum(Grau_Complex_2(C_r - C_e)) + \sum_{v_e \in C_e} min(dist(v_e, u_r) \quad \forall u_r \in C_r)}{|C_r \cap C_e|} \tag{4}$$

5. Metodologia

A avaliação da proposta ocorreu em uma hierarquia aleatória de estrutura irregular, onde os conceitos da hierarquia são distribuídos a partir de métodos provenientes da teoria de grafos. A hierarquia é gerada em quatro etapas. A primeira calcula um grafo aleatório, onde as relações entre os elementos são baseadas na distância euclidiana, um exemplo de

grafo aleatório pode ser visto na Figura 4. A segunda etapa encontra o elemento mais importante, com a finalidade de identificar o conceito mais complexo. Este conceito é o elemento do grafo com maior valor de centralidade *betweenness*[5], um exemplo de grafo com valores de centralidade é mostrado na Figura 5. A terceira etapa encontra uma árvore, onde a raiz é o elemento mais importante. O método para este objetivo busca uma árvore geradora mínima[6], um exemplo de árvore geradora minimia pode ser vista na Figura 6. A quarta etapa gera a hierarquia, esse processo direciona as arestas da árvore para a raiz.

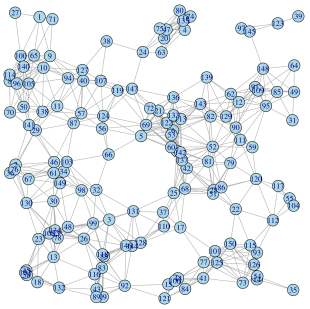

Figura 4. Grafo aleatório

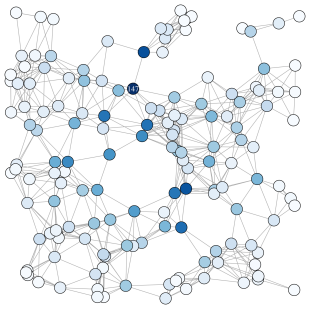

Figura 5. Centralidade do vértice

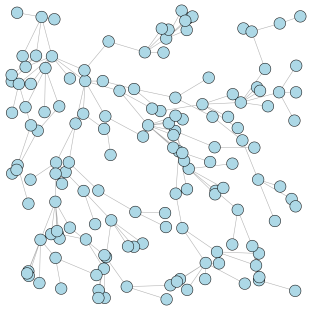

Figura 6. Árvore geradora mínima

Na hierarquia gerada os elementos são os conceitos e as arestas direcionadas representam as dependências de um conceito menos para um conceito mais complexo, um exemplo de hierarquia pode ser visto na Figura 7.

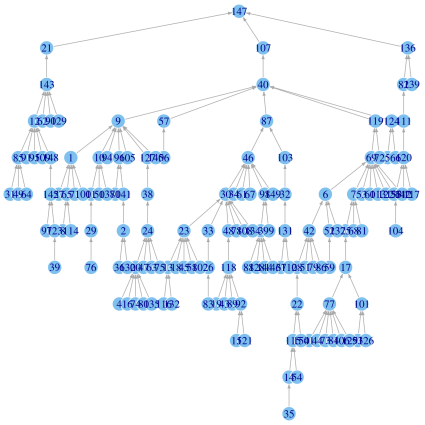

Figura 7. Hierarquia aleatória

O algoritmo utilizado para gerar a hierarquia está em **Algorithm 1**. A primeira função, linha 1, retorna, para G, um grafo aleatório com 150 vértices. A função gera 150 vértices sobre um plano cartesiano seguindo uma distribuição uniforme com valores entre 0 e 1 e as arestas são adicionadas caso a distância euclidiana entre os vértices seja menor que 0.15. Para segunda e a terceira função, linha 2 e 3, utilizamos o pacote *igraph*[7]

[5] http://igraph.org/r/doc/betweenness.html
[6] http://igraph.org/r/doc/mst.html
[7] http://igraph.org/r/

para a linguagem de programação *R*. A última função, linha 4, direciona as arestas da árvore encontrada anteriormente para os caminhos convergirem para a raiz, o conceito mais complexo.

Algorithm 1 Gerar hierarquia aleatória

1: $G \leftarrow grafo_aleatorio(150, 0, 1, 0.15)$
2: $raiz \leftarrow maior(betweenness(G))$
3: $T \leftarrow arvore_geradora_minima(G)$
4: $H \leftarrow arvore_direcionada(T, raiz)$

5.1. Métricas

O algoritmo ABC foi executado na hierarquia de conceitos definida na seção anterior. O parâmetro de entrada é o conjunto de conceitos da questão de entrada. O parâmetro de saída é o conjunto de conceitos da questão resultante da iteração. O número da população de abelhas foi de 80 e o número de questões geradas foi 60. Os resultados obtidos na simulação são avaliados de acordo com a complexidade das questões selecionadas e a proximidade dos conceitos avaliados. A avaliação ideal proposta pelo trabalho tem nível de complexidade crescente das questões e avalia-se conceitos próximos.

6. Resultados da avaliação

6.1. Resultados da simulação da fitness 1

O resultado do grau de complexidade por questão pode ser visto na Figura 8. Assim como esperado o grau de complexidade das questões tem um comportamento crescente, isso mostra que o nível das questões aumenta conforme as novas questões são geradas. A hierarquia com os vértices coloridos por mapa de calor, mensura quantas vezes o conceito foi escolhido durante as 60 iterações do algoritmo. A hierarquia colorada por mapa de calor pode ser visto na Figura 9, os vértices com coloração próxima à cor verde foram escolhidos poucas vezes durante as 60 iterações, ao passo que os vértices próximos à cor vermelha foram escolhidos muitas vezes. Pela coloração percebe-se que a base da hierarquia contém vértices amarelados e o centro da hierarquia contém alguns vermelhos, isso mostra que os vértices folhas e próximos a eles foram mais requisitados.

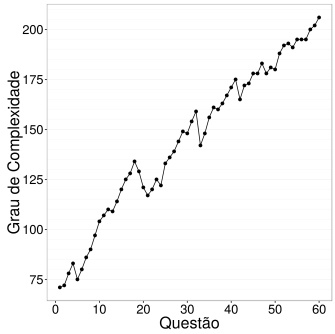

Figura 8. Grau de Complexidade por questão

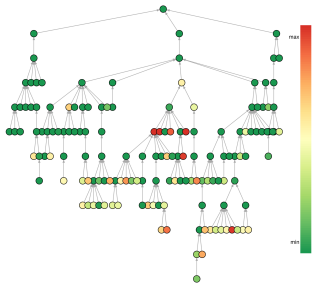

Figura 9. Hierarquia colorada por mapa de calor

O tempo de execução da simulação utilizando a fitness 1 foi menos de 1 hora para gerar uma avaliação de 60 perguntas. Onde cada pergunta consiste em um conjunto de

conceitos. Os conceitos de entrada para o algoritmo ABC foram os 7 conceitos apresentados na Figura 10. Após a execução do algoritmo ABC os conceitos resultantes são apresentados na Figura 11 e após a execução sequencial de 60 vezes do algoritmo os conceitos resultantes podem ser vistos na Figura 12. A área hachurada demonstra que ao final das iterações os conceitos foram distribuídos para as laterais e em direção a base da hierarquia. A fitness 1 conseguiu aumentar o grau de complexidade das questões em sequência, porém não escolheu conceitos próximos aos iniciais e nem conceitos acima dos iniciais.

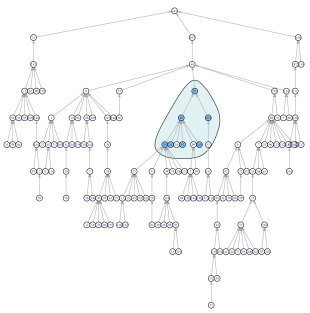

Figura 10. Conceitos de entrada para o ABC

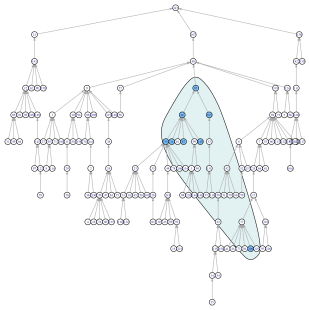

Figura 11. Conceitos resultantes da questão 1

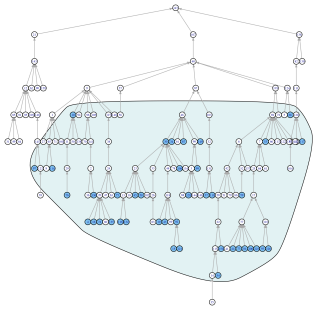

Figura 12. Conceitos resultantes da questão 60

6.2. Resultados da simulação da fitness 2

O resultado do grau de complexidade da fitness 2 por questão pode ser visto na Figura 13. O grau de complexidade da fitness 2 tem um comportamento crescente, apesar da queda no final. A hierarquia com os vértices coloridos por mapa de calor, pode ser visto na Figura 14. Percebe-se que os vértices centrais são mais requisitados, pois contém uma coloração avermelhada e existem poucos vértices amarelados em volta da região central. Desta maneira a fitness 2 conseguiu manter o comportamento crescente do grau de complexidade das questões e também conseguiu manter mais próximos os conceitos escolhidos aos conceitos iniciais, em relação a fitness 1.

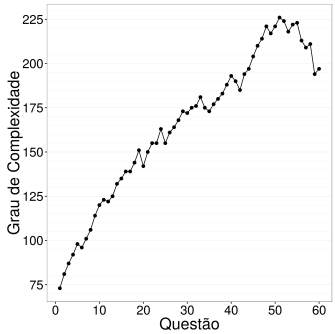

Figura 13. Grau de Complexidade por questão

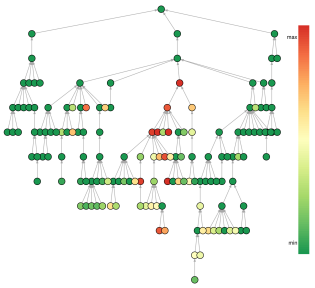

Figura 14. Hierarquia colorada por mapa de calor

A diferença dos resultados da fitness 1 para a fitness 2 é o tempo de execução. O tempo de execução da fitness 2 é perto de 5 horas. Essa diferença é significante, pois

a complexidade computacional do cálculo da fitness 2 é maior que a complexidade da fitness 1. Analisando as áreas hachuradas das sequencias de conceitos resultantes nas Figuras 15, 16 e 17, percebemos que os conceitos escolhidos pelo algoritmo são mais próximos, lateralmente, aos conceitos iniciais e também apresentam uma distribuição em direção da base da hierarquia.

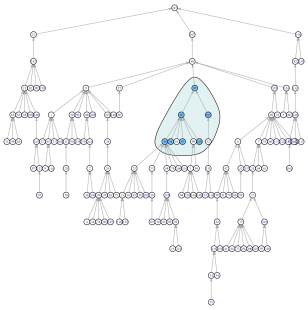

Figura 15. Conceitos de entrada para o ABC

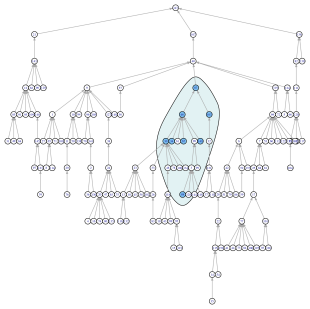

Figura 16. Conceitos resultantes da questão 1

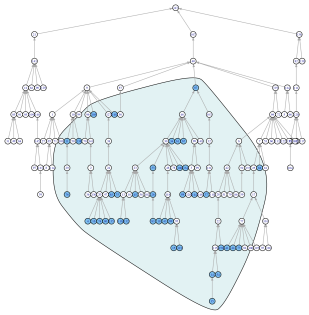

Figura 17. Conceitos resultantes da questão 60

7. Conclusão

O presente trabalho realizou um estudo empírico sobre o tema avaliação mediada por ambientes computacionais. Como resultado dessa analise gerou-se uma heurística para criação automática de avaliações. A próxima questão é escolhida utilizando o algoritmo de inteligência coletiva chamado ABC, informações como os conceitos que a questão avalia e a grau de dificuldade da questão são utilizados para definir a função *fitness*.

Em cenários onde a base de conhecimento e questões é pequena a busca exaustiva pela próxima questão da avaliação é vantajosa, mas quando analisada uma base com conteúdos e questões com tamanho considerável identifica-se a necessidade da utilização de uma nova metodologia para seleção da próxima pergunta. Sendo assim, propomos uma modelagem de conhecimentos e de questões para experimentação. A modelagem de conhecimentos deu-se através de uma hierarquia de conceitos, onde os nós da hierarquia são os conceitos e as relações de dependência de aprendizado entre os conceitos são estipuladas pelas arestas. As questões avaliam os conceitos da hierarquia e possuem um grau de complexidade diferente estipulado para cada função *fitness*.

As pesquisas realizadas indicam a necessidade da criação de uma base de dados unificada, contendo os conhecimentos e competências dos domínios ensinados. Outra necessidade é a disponibilização do banco de questões das provas que avaliam o conhecimento de estudantes das várias modalidades de ensino nacionalmente. As métricas e os experimentos criados geraram aleatoriamente uma base de conhecimentos e questões que avaliam esses conhecimentos, para em um segundo momento suprir as necessidades do banco de conhecimentos e questões.

Referências

[Almond et al. 2007] Almond, R. G., DiBello, L. V., Moulder, B., and Zapata-Rivera, J.-D. (2007). Modeling diagnostic assessments with bayesian networks. *Journal of Educational Measurement*, 44(4):341–359.

[Augustus 2009] Augustus, G. (2009). Inteligência de enxame e o algoritmo das abelhas. `http://www.ime.usp.br/~gold/cursos/2009/mac5758/1512/Glaucus.pdf/`. acessado em 02/12/2015.

[Demo 2002] Demo, P. (2002). *Mitologias da avaliação: de como ignorar, em vez de enfrentar problemas.* São Paulo: Editora Autores Associados, 2 edition.

[Desmarais and Baker 2011] Desmarais, M. C. and Baker, R. S. J. d. (2011). A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, 22(1):9–38.

[Driscoll 2005] Driscoll, M. P. (2005). *Psychology of Learning for Instruction.* Florida State University: Pearson, 3 edition.

[Hoffmann 2001] Hoffmann, J. (2001). *Avaliação: mito e desafio: uma perspectiva construtivista.* Porto Alegre: Editora Mediação, 44 edition.

[Karaboga and Akay 2009] Karaboga, D. and Akay, B. (2009). A survey: algorithms simulating bee swarm intelligence. *Artificial Intelligence Review*, 31(1-4):61–85.

[Leão and Reis 2006] Leão, P. L. C. and Reis, C. A. L. (2006). Uma linguagem visual para avaliação adaptativa de aprendizagem baseada em gramática de grafos. *RENOTE*, 4(2).

[Magalhães and Carpinteiro 2004] Magalhães, D. R. and Carpinteiro, O. A. S. (March 14 - 17, 2004). Um modelo computacional de avaliação de alunos. In *World Congress on Engineering and Technology Education*, pages 658–661, São Paulo.

[Morais 2002] Morais, C. T. Q. (2002). Avaliação interativa de alunos em curso na web baseado em automatos finitos. Dissertação de mestrado, Instituto de Informática. Universidade Federal do Rio Grande do Sul., Porto Alegre.

[Pimentel 2006] Pimentel, E. (2006). Um modelo para avaliação e acompanhamento continuo do nível de aquisição de conhecimentos do aprendiz. Tese de doutorado, Instituto Tecnológico de Aeronáutica, São José dos Campos.

[Serapião 2009] Serapião, A. B. d. S. (2009). Fundamentos de otimização por inteligência de enxames: uma visão geral. *Sba: Controle Automação Sociedade Brasileira de Automática*, 20.

[Taveira 2003] Taveira, R. D. (2003). Implementando um mecanismo de processamento de conhecimento em uma ferramenta de ead. Dissertação de mestrado, Universidade Estadual do Ceará, Fortaleza.

[Tesseroli 2015] Tesseroli, R. d. C. (2015). Arcabouço para mapeamento do conhecimento do aluno baseado no modelo de redescrição representacional. Dissertação de mestrado, Universidade Federal do Paraná, Curitiba.

[Zanella 2005] Zanella, R. (2005). Easy - sistema de avaliação via web baseado no hyperautomaton. Dissertação de mestrado, Instituto de Informática. Universidade Federal do Rio Grande do Sul., Porto Alegre.