

# Titulo

**Carlos R. Lopes<sup>1</sup>, Augusto B. Branquinho<sup>1</sup>, Fabiano A. Dorça<sup>1</sup>  
Márcia A. Fernandes<sup>1</sup>, Renan G. Cattelan<sup>1</sup>**

<sup>1</sup>Faculdade de Computação – Universidade Federal de Uberlândia (UFU)  
Campus Santa Mônica – Bloco 1B – Sala 1B148  
Av. João Naves de Ávila, 2.121 – Bairro Santa Mônica  
CEP 38400-902 – Uberlândia-MG – Brasil

{crlopes, augusto, fabiano, marcia, renan}@facom.ufu.br

*Abstract. texto.*

*Resumo. texto.*

## 1. Introdução

Computação ubíqua é conhecida como a terceira onda da computação, a qual visa tornar os dispositivos computacionais mais integrados à vida das pessoas. Ela tem ajudado as pessoas a realizarem suas atividades diárias em diferentes domínios, especialmente no domínio educacional. Tal tecnologia auxilia instrutores e alunos durante as suas atividades acadêmicas conhecimento em qualquer lugar e a qualquer momento, fazendo uso de formas não intrusivas para recuperar o conteúdo desejado. Captura e acesso multimídia e ciência de contexto são dois importantes temas em pesquisas de computação ubíqua. Enquanto o primeiro produz automaticamente um grande volume de mídias digitais (ex., vídeo, áudio, slides, comentários de texto), o segundo oferece meios adequados para integrar e acessar tal conteúdo. Ao fundir os dois conceitos, os usuários podem se beneficiar de ferramentas que lhes permitam recuperar, visualizar e interagir com os artefatos de mídia capturados.

Neste artigo o uso da computação ubíqua se faz presente no domínio educacional. Uma arquitetura de acesso contextual foi desenvolvida para personalização e recomendação de conteúdo multimídia interativo capturado em ambientes educacionais instrumentados. Essa arquitetura utiliza uma abordagem que considera informações de contexto, preferências do usuário e restrições de apresentação a fim de personalizar a experiência de acesso, adequando-a às necessidades dos usuários. Como estudo de caso, as propostas foram implementadas no *Classroom eXperience*, que é uma plataforma de computação ubíqua para captura multimídia concebida para gravar automaticamente as aulas ministradas e disponibilizá-las posteriormente aos alunos.

Um ambiente de computação ubíqua, tal qual o *Classroom eXperience*, deve ser estendido para incluir maiores funcionalidades de personalização do conteúdo a ser apresentado ao aluno. Dado que se tem um volume enorme de mídias, que em essência são objetos de aprendizagem, torna-se útil ter uma ferramenta que possa fornecer ao aluno objetos de aprendizagem que irão propiciar-lhe atingir determinadas competências. Para isto será necessário um mecanismo que possa gerar uma sequência de materiais educacionais que, uma vez assimilados, irão fazer com que a competência desejada seja alcançada.

O trabalho descrito neste artigo mostra o processo de planejamento de uma sequência de conteúdos para atingir um determinado objetivo educacional. Para isto considera-se que os conteúdos serão representados na forma de objetos de aprendizagem. Os objetivos educacionais ou competências a serem atingidas também serão descritos por meio de objetos de aprendizagens. Desta forma ter-se-á um conjunto de objetos de aprendizagem que deverão ser dispostos numa ordem de apresentação que leve em consideração determinadas restrições. Por exemplo, um objeto de aprendizagem não pode ser apresentado sem que outro, considerado como seu pré-requisito tenha sido apresentado primeiramente. Para isto um primeiro esforço para a solução do problema foi feita com algoritmos genéticos. Para isto são levados em consideração o perfil do estudante, caracterizado pelo seu estilo de aprendizagem e estado cognitivo, como forma de garantir uma personalização do conteúdo.

O artigo está organizado como descrito a seguir. A seção xx apresenta a plataforma Classroom eXperience de computação ubíqua com foco em educação. A seção xx descreve o mecanismo para diagnóstico do estilo de aprendizagem do aluno necessário para o sequenciamento de objetos de aprendizagem. O processo de sequenciamento do conteúdo usando algoritmo genético é apresentado na seção xx. Na seção xx descreve-se uma conclusão do trabalho apresentado.

## **2. A Plataforma Classroom eXperience (CX)**

A plataforma CX é uma plataforma de captura multimídia concebida para gravar automaticamente as aulas ministradas e disponibilizá-las posteriormente aos alunos. Com o CX é possível capturar, armazenar e acessar o conteúdo multimídia capturado em salas de aula instrumentadas com lousas eletrônicas, microfones, câmeras de vídeo e projetores. Os fluxos de mídia de cada dispositivo são capturados e sincronizados, gerando documentos hipermídia em diferentes formatos de apresentação. Construído sobre a plataforma iClass (Pimentel et al., 2007), o CX complementa seu alicerce implementando a camada CAL, como meio de comunicação, e também a arquitetura CAA, para personalização e recomendação de conteúdo. Tal abordagem permite a personalização da apresentação do conteúdo capturado de acordo com as preferências e o contexto de acesso de cada estudante.

Sem alterar a dinâmica convencional das aulas, o CX utiliza recursos de computação ubíqua para auxiliar tanto instrutores no processo de captura de aulas quanto estudantes no processo de acesso ao conteúdo ministrado. O CX conta com componentes de software especializados para gravar as atividades educacionais, sincronizar os diferentes fluxos de mídia gerados e disponibilizar o conteúdo através de uma interface amigável. Todos os componentes foram desenvolvidos utilizando a linguagem Java com suporte a bibliotecas Web. Além disso, as informações da aplicação bem como o conteúdo gerado são armazenados no banco de dados PostgreSQL.

## **3. Estilos de Aprendizagem**

## **4. Objetos de Aprendizagem e Sequenciamento**

Neste trabalho considera-se que objetos de aprendizagem podem representar competências a serem alcançadas. Dado que geralmente teremos um conjunto de competências a serem adquiridas, isto significa que será necessário que o usuário trabalhe

com um conjunto de objetos de aprendizagem. Tem-se o então o chamado problema do sequenciamento dos objetos de aprendizagem. Dado uma sequência aleatória de objetos de aprendizagem o problema consiste em determinar uma sequência ordenada dos mesmos levando-se em conta as restrições existentes. O uso de objetos de aprendizagem para representar competências e as possíveis restrições existentes entre as mesmas, bem como a técnica usada para resolver o problema do sequenciamento são apresentados nas subseções seguintes.

#### **4.1. Competências**

O trabalho desenvolvido foi estabelecido levando-se em conta as competências a serem adquiridas pelo usuário. Segundo xxxxx (xxxx) as competências podem ser descritas como "entidades multidimensionais constituídas por conhecimento, habilidades e fatores psicológicos que surgem de forma integrada como um comportamento em função de uma interação com o ambiente".

Existem diversas iniciativas objetivando padronizar a definição de competência de tal modo que ela poderia ser máquinas. Por exemplo, o comitê IEEE-LSTC define que padrão para uma tecnologia de aprendizagem corresponde a um padrão para definição de competências que podem ser reusadas. Ainda de acordo com a nomenclatura IEEE, um registro de competência é chamado de Definição de Competência Reusável. A definição de uma competência reusável pode ser feita, num nível mínimo, pela sequência de dois objetos de aprendizagens: um estabelecendo objetivos educacionais a serem alcançados e outro correspondendo um pré-requisito a ser satisfeito. Informações especificando objetivos, pré-requisitos e outros dados de interesse são denominados de metadados dos objetos de aprendizagem e devem aparecer na especificação dos objetos. No trabalho que foi desenvolvido adotou-se o padrão LOM (Learning Object Metadata) para especificação de metadados.

O padrão é caracterizado pela especificação de uma série de registros que descrevem os metadados. ...

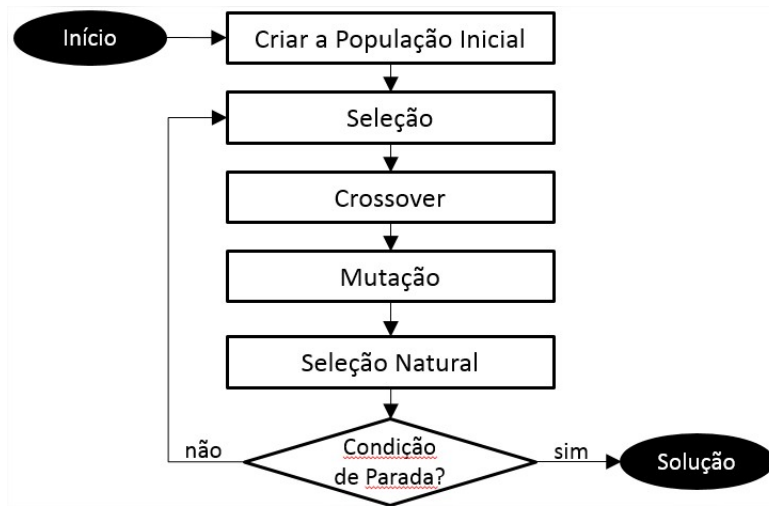
Em função do exposto tem-se que o objetivo a ser alcançado é fazer com que o usuário assimile um conjunto de competências. Assim sendo teremos um conjunto de objetos de aprendizagem não ordenados que correspondem às competências desejadas. Um cuidado deve ser tomado ao se apresentar os objetos de aprendizagem aos usuários. Não é conveniente apresentar um determinado objeto de aprendizagem sem antes de apresentar outro(s) que correspondam aos seus pré-requisitos. Desta forma faz-se necessária uma correta ordenação ou sequenciamento dos objetos, que é a solução para o problema do sequenciamento de objetos.

#### **4.2. Estratégias Evolucionárias**

Para realizar o planejamento sequencial dos objetos de aprendizado foram utilizadas estratégias evolucionárias. As estratégias têm como base o processo de evoluir as soluções com base em gerações.

Estes processos consistem em um conjunto de passos. Inicialmente é realizada a geração aleatória da população inicial. A população consiste em um grupo de indivíduos. O tamanho da população pode variar de acordo com os experimentos. Cada indivíduo

corresponde a uma possível solução de um dado problema. Após gerar a população inicial são executados métodos de evolução. A evolução, também chamada de geração, consiste primeiramente na seleção de indivíduos da população. São selecionados pares de indivíduos, também chamados de pais, e utilizando o processo de crossover e mutação são gerados novos indivíduos, também chamados filhos. Após a geração das novas soluções é realizada a seleção natural. Este procedimento é responsável por eliminar as piores soluções, mantendo a população sempre com o mesmo tamanho e com os melhores indivíduos. A Figura 1 mostra o passo a passo de um estratégia evolucionária.



**Figura 1. Passos das estratégias evolucionárias.**

Para o problema dos objetos de aprendizado, cada indivíduo corresponde a uma sequência de objetos de aprendizado ordenados de acordo com as restrições e competências. Muitos indivíduos possuem restrições que podem não serem satisfeitas pela sequência de objetos. Deste modo, o objetivo é que as estratégias encontrem as sequências que satisfaçam todas as restrições.

Para avaliar o quão bom é um indivíduo é realizado o cálculo da aptidão. Essa aptidão, comumente chamada de *fitness*, tem como base as restrições e a posição de cada objeto de aprendizado. O *fitness* do trabalho é expresso pela Equação 1 [García et al. 2010]. Nesta equação a variável  $s$  corresponde a uma sequência de objetos e  $pr(S_i)$  é o total de requisitos não satisfeitos para o objeto na posição  $i$ .

$$fitness(s) = \sum_{i=1}^n i * pr(s_i) \quad (1)$$

A Tabela 1 mostra uma sequência com 6 objetos e seus respectivos requisitos e competências. Cada competência foi representada por uma letra. Neste caso o objeto LO<sub>1</sub> não possui nenhum requisito. Já LO<sub>2</sub> tem seu requisito satisfeito pelo resultado de LO<sub>1</sub>. Em seguida, LO<sub>3</sub> possui os requisitos E e F que não são satisfeitos. Este processo é realizado para cada objeto e ao final a aptidão é dada por  $fitness(s) = 1 * 0 + 2 * 0 + 3 * 2 + 4 * 0 + 5 * 1 + 6 * 0 = 11$ .

Este trabalho utiliza duas abordagens evolucionárias, sendo o algoritmo genético

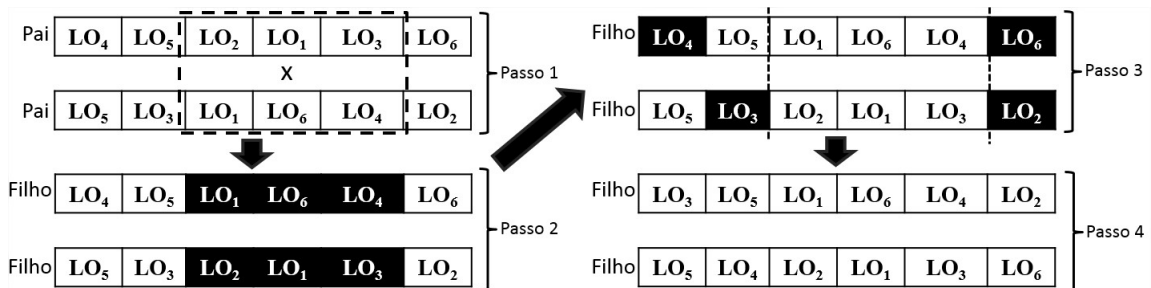
**Tabela 1. Sequência de objetos com as respectivas restrições e competências.**

| Sequência de Objetos | LO <sub>1</sub> | LO <sub>2</sub> | LO <sub>3</sub> | LO <sub>4</sub> | LO <sub>5</sub> | LO <sub>6</sub> |
|----------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Requisitos           |                 | B               | C, E, F         | C, D            | F, K            | G               |
| Competências         | B               | C, D            | G               | F               | H               | K               |

tradicional e o algoritmo *steady state* [Luke 2013]. A diferença fundamental destes procedimentos é que enquanto o algoritmo genético seleciona vários pares de indivíduos durante uma evolução o *steady state* escolhe apenas um par. O objetivo do *steady state* é que ele exerça uma pressão seletiva maior [Luke 2013]. Outro ponto que diverge entre estes métodos é que o algoritmo genético possui como parâmetro de parada um número fixo de gerações e o *steady state* para de executar após um número de iterações sem evolução.

Para realizar a seleção dos indivíduos que sofreram crossover foram usadas três técnicas, sendo por roleta, torneio e aleatório. A roleta realiza uma seleção de acordo com um sorteio considerando o percentual de quão bom são os indivíduos. Já o torneio necessita de um parâmetro inicial  $n$  que determina quanto elementos serão inicialmente selecionados através da roleta. Uma vez escolhidos os  $n$  indivíduos o melhor dentre eles é dado como resultado do torneio. Por último, o método aleatório apenas escolhe de forma aleatório os indivíduos.

Existem diversos métodos de crossover, sendo que neste trabalho foi usado o PMX. O crossover PMX consiste em selecionar uma região de cada indivíduo e posteriormente é realizada a troca dos objetos entre dentro do intervalo da região selecionada. A Figura 4 exibe o crossover PMX.



**Figura 2. Passo a passo do crossover PMX.**

A partir de dois indivíduos, chamados de pais, são gerados dois novos filhos. No primeiro passo é selecionado um intervalo de objetos de cada pai. No segundo passo é realizada a troca dos objetos dentro do intervalo. Posteriormente, no passo 3 são verificados quais objetos se repetem para cada indivíduo. Ao final é realizada mais uma troca dos objetos repetidos seguindo a mesma sequência em que ocorreram as repetições.

Cada indivíduo possui uma probabilidade de ocorrer mutação. Essa probabilidade é definida no início da execução. Neste trabalho a mutação realiza uma simples troca de posição de um objeto. Após a mutação todos os novos indivíduos são adicionados a população atual. Posteriormente, a seleção natural remove os piores indivíduos com o objetivo de manter a população com o mesmo tamanho inicial. Ao final, se a solução ótima foi encontrada ou o critério de parada foi satisfeito o melhor indivíduo é dado como solução. É importante lembrar que o critério de parada para o algoritmo genético

e o *steady state* são, respectivamente, o número de gerações e o limite de iterações sem melhoria da solução.

### 4.3. Algoritmos de Busca

Os algoritmos de busca têm como meta a partir de uma estado inicial realizar um conjunto de passos até encontrar uma solução. No processo de busca temos o estado atual que está sendo explorado e a partir dele são realizadas expansões

## 5. Resultados

Para avaliar a aplicação das estratégias evolucionárias e algoritmos de busca foram realizados vários experimentos com o objetivo de determinar quais métodos encontram os melhores resultados. Para cada experimento foram executadas 100 vezes e o resultado final é dado pela média destes resultados. Foram utilizados 9 diferentes conjuntos de objetos, sendo que cada grupo possui respectivamente as seguintes quantidades de 10, 20, 30, 40, 50, 60, 70, 80 e 90 objetos. Estes objetos foram gerados aleatoriamente com restrições e competência variando de 0 a 4.

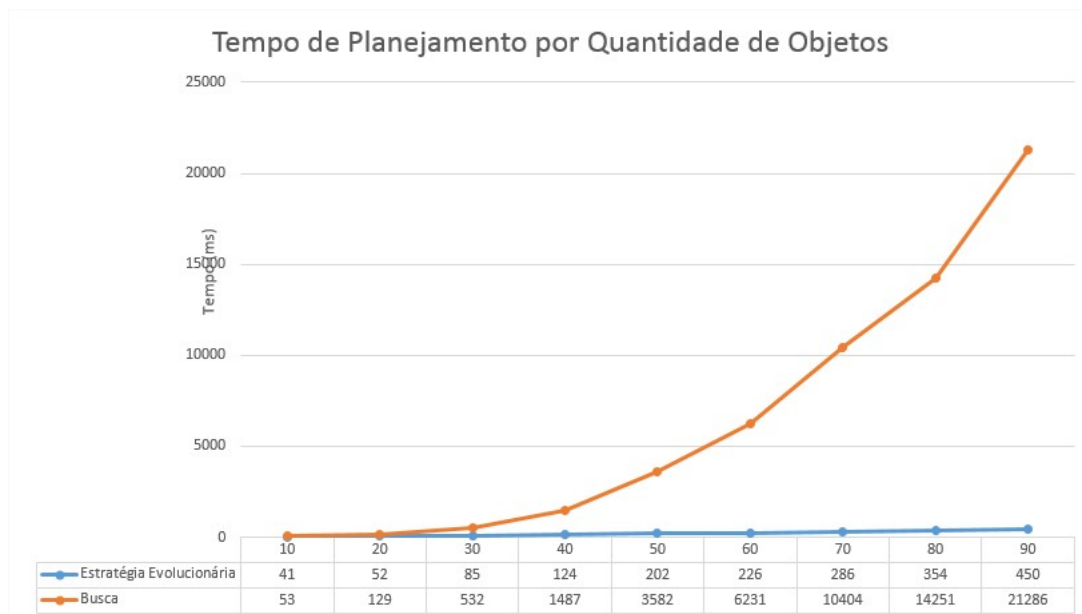
Os experimentos foram separados em 3 diferentes grupos. Em cada grupo foram alterados os parâmetros com o objetivo de buscar os melhores resultados.

O primeiro grupo utilizou algoritmo genético tradicional com população contendo 50 indivíduos. O critério de parada consiste em encontrar o *fitness* ótimo (igual a 0) ou se ultrapassar o limite de gerações. O máximo de gerações estipuladas foram de 100 e 200. Para o processo de seleção foram usados os métodos de roleta, aleatório e torneio 3. Foi usada uma mutação simples com probabilidades de 4%, 10% e 20%.

Já no segundo grupo foi usado o algoritmo *steady state*. O critério de parada ocorre se a busca encontrar a solução ótimo ou se for alcançado o limite de gerações sem melhoria da solução.



Figura 3. *Fitness* de acordo com a quantidade de objetos de aprendizagem.



**Figura 4. Tempo para a execu  o dos experimentos de acordo com a quantidade de objetos de aprendizagem.**

## 6. Conclus  o

## 7. Refer  ncias Bibliogr  ficas

### Refer  ncias

Garc  a, A., de Marcos, L., Garc  a, E., and Guti  rrez, J. A. (2010). Courseware sequencing using heuristic and local search. In Arabnia, H. R., Bahrami, A., and Solo, A. M. G., editors, *CSREA EEE*, pages 70–75. CSREA Press.

Luke, S. (2013). *Essentials of Metaheuristics*. 2 edition.