

**PROGRAMAÇÃO DE CURSOS UNIVERSITÁRIOS BASEADA EM CURRÍCULOS
VIA UMA META-HEURÍSTICA HÍBRIDA GRASP-ILS UTILIZANDO O
PROCEDIMENTO KEMPESWAP**

Saulo Henrique D'Carlos Barbosa

Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Programa de Pós-Graduação em Modelagem Matemática e Computacional
Av. Amazonas, 7675 - CEP 30510-000, Belo Horizonte - MG
saulo_comp@yahoo.com.br

Sergio Ricardo de Souza

Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Programa de Pós-Graduação em Modelagem Matemática e Computacional
Av. Amazonas, 7675 - CEP 30510-000, Belo Horizonte - MG
sergio@dppg.cefetmg.br

Marcone Jamilson Freitas Souza

Departamento de Computação
Universidade Federal de Ouro Preto
35.400-000 Ouro Preto, MG
marcone@iceb.ufop.br

RESUMO. Este trabalho trata o Problema de Programação de Cursos Universitários baseada em Currículos. Este problema consiste em alocar um período e uma sala para as aulas de todas as disciplinas, evitando conflitos entre disciplinas de um mesmo currículo. São consideradas as características e instâncias da Competição Internacional de Programação de Horários (International Timetabling Competition - ITC2007). Sendo o problema da classe NP-difícil, propõe-se um algoritmo heurístico, nomeado GRASP-ILS, que combina os procedimentos heurísticos GRASP, Iterated Local Search e Kempeswap. O algoritmo proposto foi comparado com os cinco melhores algoritmos da competição, assim como com o algoritmo sem o procedimento Kempeswap. Os resultados computacionais obtidos mostraram a influência positiva do procedimento Kempeswap no algoritmo, assim como permitiram validar sua eficiência.

Palavras-Chave: Programação de cursos universitários baseada em currículos, Meta-Heurísticas, Kempeswap, ITC 2007.

ABSTRACT. This paper addresses the Curriculum Based Course Timetabling Problem. This problem consists in allocating a timeslot and room for the classes of all disciplines, avoiding conflicts between courses of the same curriculum. We considered the characteristics and instances of the International Competition timetabling - ITC2007. Since the problem class is NP-hard, we propose a heuristic algorithm, named GRASP-ILS, which combines the heuristic procedures GRASP, Iterated Local Search and Kempeswap. The proposed algorithm was compared with the best five algorithms in the competition, as well as with the algorithm without the procedure Kempeswap. The computational results showed the positive influence of Kempeswap procedure in the algorithm, as well as allowed to validate its efficiency.

Keywords: Curriculum Based Course Timetabling Problem, Meta-heuristics, Kempeswap, ITC 2007.

1. INTRODUÇÃO

De acordo com Burke e Petrovic (2002), a classe de problemas *Timetabling* (Problema de Programação de Horários - PPH) tem atraído a atenção da comunidade científica nos últimos quarenta anos, devido a sua importância no cotidiano e complexidade de resolução (NP-Completo). Segundo Willemen (2002), o problema abrange todas as atividades que envolvem o processo de construção de quadros de horários, sendo o PPH de grande aplicação nas mais variadas áreas, tais como nas indústrias, hospitais, empresas de transporte, eventos esportivos, até escolas e universidades. Em essência, o problema consiste na atribuição de um número de eventos, cada um com uma série de características, a um número limitado de recursos, sujeito a determinadas restrições (QU, 2002).

O PPH apresenta como uma de suas principais variações o Problema de Programação de Horários Educacionais (PPHE). Um importante marco para a classe de problemas do tipo PPHE foi a realização da competição internacional de programação de horários (*International Timetabling Competition - ITC 2002* e *ITC 2007*). Nessa competição, o objetivo foi o de apresentar diferentes técnicas para resolução de problemas de PPHE e compará-las. A importância do evento deve-se ao fato de não ser comum autores realizarem testes computacionais e comparações entre diferentes técnicas, devido a diversidade de características envolvidas em cada abordagem. Em sua última edição, o *ITC 2007* apresentou três categorias de problemas: Programação de Exames, Programação de Cursos Pós-Matrículas e Programação de Cursos baseada em Currículos (*Curriculum based Course Timetabling - CBCT*).

O CBCT, objeto de estudo neste trabalho, é um problema geralmente encontrado em universidades, que apresentam, como característica, a possibilidade de o aluno escolher quais disciplinas serão cursadas a cada semestre, sendo as possíveis disciplinas limitadas pelo currículo ao qual o aluno pertence. Na variação estudada neste artigo, busca-se alocar um período e uma sala para cada aula de todas as disciplinas, levando-se em conta os conflitos entre disciplinas de um mesmo currículo. O presente artigo utiliza não só das instâncias fornecidas pelo *site* do evento, bem como respeita as regras da competição para a realização dos experimentos e comparações.

Segundo Lewis (2008), dentre uma grande variedade de abordagens (métodos sequenciais, métodos baseados em restrições, etc.) encontradas para a resolução do PPHE tem se destacado a utilização de técnicas meta-heurísticas. Como se trata de um problema da classe NP-difícil, a utilização de meta-heurísticas para sua resolução se justifica, por apresentar boas soluções sem um elevado custo computacional. Vários trabalhos relevantes que relatam a resolução do PPH via meta-heurísticas podem ser encontrados na literatura, dentre eles: Lü e Hao (2010), que utiliza Busca Tabu com aperfeiçoamentos como a utilização de perturbações e movimentos avançados de busca; Muller (2008), que utiliza *Simulated Annealing* aliado a outros métodos de busca local e análise estatística; Souza (2000), que realiza o teste de várias meta-heurísticas para resolução do problema; Bai *et al.* (2008), que utiliza *Simulated Annealing* com aperfeiçoamentos; e Barbosa *et al.* (2010), que utiliza uma implementação híbrida GRASP e Busca Tabu. Para uma completa revisão do estado da arte para o PPHE veja Schaerf (1999), Burke e Petrovic (2002) e Lewis (2008).

Este artigo apresenta um algoritmo híbrido, nomeado GRASP-ILS, baseado nas meta-heurísticas *Greedy Randomized Adaptive Search Procedure* - GRASP (FEO e RESENDE, 1995) e *Iterated Local Search* - ILS (LOURENÇO *et al.*, 2003) para resolução do problema estudado. O algoritmo também aplica um procedimento de exploração do espaço de soluções denominado *Kempeswap* (LÜ e HAO, 2010), em que um conjunto de aulas que definem uma cadeia Kempe são alteradas. A utilização deste procedimento na técnica criada representa um aperfeiçoamento em busca de melhores resultados e maior eficiência.

O restante deste trabalho está organizado como segue. Na próxima seção, o problema de programação de disciplinas universitárias baseada em currículos é discutido e detalhado. Na seção 3, é apresentada a metodologia utilizada para aplicação das técnicas pesquisadas. Os resultados obtidos são apresentados na seção 4, juntamente com as comparações com os resultados da literatura. Por fim, são apresentadas as conclusões e os trabalhos futuros.

2. DESCRIÇÃO DO PROBLEMA ESTUDADO

O problema de programação de cursos universitários baseada em currículos (*Curriculum based Course Timetabling* - CBCT) é uma variação do PPHE, descrita na terceira modalidade de problema da competição *International Timetabling Competition 2007* (<http://www.cs.qub.ac.uk/itc2007/>).

De acordo com Di Gaspero e Schaerf (2007), o problema consiste em realizar a programação semanal de disciplinas de vários cursos universitários dentro de um determinado número de períodos de tempo e salas disponíveis. Os conflitos entre as disciplinas são determinados pelos currículos em que as mesmas estão envolvidas, além de outras restrições que determinam a viabilidade e qualidade de um quadro de horários construído.

De acordo com De Cesco *et al.* (2010), as principais entidades envolvidas no problema são:

- Dias, Horários e Períodos: dado um número de dias letivos, cada dia é dividido em um dado número de períodos, sendo um horário correspondente ao par dia/período;
- Currículos: um currículo é composto por um grupo de disciplinas que possam ter estudantes em comum, e, por isso, as disciplinas de um mesmo currículo não devem ser agendadas em um mesmo período;
- Disciplinas e Professores; cada disciplina apresenta um professor alocado para lecionar (previamente alocado pela instituição); o número de aulas; o número de alunos; o número mínimo de dias exigido na alocação; e, também, um conjunto de horários em que a mesma não está disponível para alocação;
- Salas: cada sala apresenta sua respectiva capacidade;

Uma solução para o problema é dada pela atribuição de um período e uma sala para todas as aulas de todas as disciplinas.

A avaliação de uma solução é feita tendo em vista: i) as restrições fortes, que são essenciais, pois estão ligadas à viabilidade; e ii) as restrições fracas, que são almejadas para melhorar a qualidade de uma solução, mas podem ser desrespeitadas, caso não exista outra alternativa. As restrições consideradas pelo problema são descritas em De Cesco *et al.* (2010) e apresentadas na Tabela 1.

3. METODOLOGIA PROPOSTA

3.1. Representação de uma Solução

O problema apresenta q disciplinas (d_1, \dots, d_q) , p períodos $(1, \dots, p)$ e m salas s_1, \dots, s_m . Cada disciplina d_i consiste em l_i aulas, agendadas em períodos distintos e atendendo a_i alunos. Cada sala s_j apresenta uma capacidade cap_j em termos de número de lugares disponíveis. O problema apresenta também g grupos de disciplinas, os quais são chamados currículos. As disciplinas de um currículo são agrupadas por apresentarem estudantes em comum e, assim sendo, não devem ser agendadas em um mesmo período.

Tabela 1: Restrições do problema de programação de cursos universitários baseada em currículos

RESTRIÇÕES FORTES	RESTRIÇÕES FRACAS
R01 - Restrição Aulas: Determina que todas as aulas de uma disciplina devem ser agendadas e alocadas em períodos distintos. Uma violação ocorre se alguma aula da disciplina não é agendada ou se duas aulas distintas são agendadas no mesmo período;	R05 - Restrição Capacidade da Sala: Para cada aula agendada a sala alocada deve possuir capacidade maior ou igual à exigida pela disciplina. Nesta restrição para cada estudante a mais que a capacidade da sala, têm-se uma violação;
R02 - Restrição Conflitos: As aulas de disciplinas de um mesmo currículo ou aulas de duas ou mais disciplinas que possuem o mesmo professor, não devem ser agendadas para um mesmo período. Um período com duas disciplinas conflitantes conta como uma violação. Um período com três disciplinas conflitantes conta como duas violações, sendo uma violação para cada par de disciplinas;	R06 - Restrição Mínimo de Dias: As aulas de uma disciplina devem estar agendadas em um mínimo de número dias pré estabelecido. Para cada dia a menos que o mínimo estabelecido têm-se uma violação da restrição;
R03 - Restrição Ocupação das Salas: Duas ou mais aulas distintas não podem ocupar a mesma sala em um mesmo período. Sendo considerado como o número de violações cada sala extra encontrada no período conflitante;	R07 - Restrição Aulas Isoladas: Aulas de disciplinas de um mesmo currículo devem ser agendadas sempre de maneira adjacente, ou seja, em horários consecutivos de um mesmo dia. Nesta restrição os horários alocados para um currículo de disciplinas são analisados, e cada disciplina que não é adjacente a nenhuma outra conta como uma violação;
R04 - Restrição Disponibilidade: Caso o professor que leciona a disciplina não tenha disponibilidade em determinado período, nenhuma aula da disciplina deve ser alocada no mesmo. Cada aula agendada em um período com indisponibilidade do professor representa uma violação;	R08 - Restrição Estabilidade da Sala: Todas as aulas de uma disciplina devem ocorrer em uma mesma sala. Cada sala diferente utilizada por uma disciplina, conta como uma violação;

A representação de uma solução é dada por uma matriz T de inteiros $q \times p$, na qual $T_{ik} = j$ (com $1 \leq j \leq m$) significa que a disciplina d_i tem aula na sala r_j no período k e $T_{ik} = -1$ significa que a disciplina d_i não possui nenhuma aula alocada no período k . Ou seja, uma solução é dada por uma matriz que relaciona disciplina por períodos. Esta matriz é preenchida com valores inteiros, que representam em qual sala a disciplina será ministrada. Por meio desta representação, é possível obter a grade de horário das disciplinas, bem como escalonamento de salas para as mesmas.

3.2. Vizinhança

Para exploração do espaço de soluções vários tipos de movimentos foram testados, e foram usados aqueles que tiveram maior impacto nos resultados obtidos. Após testes empíricos, foram adotados os seguintes movimentos:

- **MovimentoPeríodo:** consiste em alterar uma disciplina de um período para outro em que não haja conflito com disciplinas do mesmo currículo;
- **MovimentoSala:** uma aula de uma disciplina tem sua sala alterada para outra sala não conflitante (em termos de ocupação);
- **MovimentoAula:** uma disciplina tem uma de suas aulas selecionada, um novo período para essa aula é selecionado e, por fim, uma nova sala é escolhida. Caso haja conflitos, a aula da disciplina conflitante tem uma troca de horários entre os horários envolvidos no movimento;
- **MovimentoEstabilidadeSala:** este movimento é utilizado no intuito de diminuir o número de violações a respeito da restrição estabilidade de salas. O movimento consiste em escolher uma nova sala e atribuí-la para todos os horários de uma determinada disciplina. Caso haja conflitos de sala após a alocação, as salas das aulas das disciplinas conflitantes são trocadas pelas salas que estavam na disciplina que recebeu o movimento;
- **MovimentoCurriculoCompacto:** este movimento é utilizado no intuito de diminuir o número de violações a respeito da restrição aulas isoladas. Uma aula isolada (ou seja,

que não possui aulas do mesmo currículo alocadas de forma adjacente) é escolhida e alocada a um horário adjacente a outra aula qualquer do currículo;

- **Movimento Troca De Aulas E Salas:** neste movimento são selecionadas duas disciplinas que utilizam a mesma sala em horários diferentes. Após escolher as disciplinas envolvidas, as mesmas têm seus horários alternados, mantendo a alocação para a sala.

Os quatro primeiros movimentos são baseados nos movimentos adotados por Muller (2008). Esse autor apresenta uma abordagem ao problema por meio de uma técnica híbrida que utiliza meta-heurísticas e bases estatísticas. Já o quinto movimento é uma proposição do presente artigo. Os movimentos propostos por Muller (2008), juntamente com o quinto movimento, foram os que apresentaram maior eficiência para exploração do espaço de busca dentre os diferentes tipos de movimentos testados.

3.3. Função de Avaliação

A função de avaliação conceitua um quadro de horários analisando a qualidade e a viabilidade de uma solução encontrada. Esta análise é feita de modo a verificar as violações das restrições estabelecidas, restrições estas que são divididas em restrições fortes e restrições fracas, conforme apresentado na seção 2 deste trabalho.

O valor dado à violação de uma restrição está diretamente ligado a relevância desta no problema. Dessa forma, as restrições fortes possuem um valor de peso alto, para que sejam prioritárias (no caso, elas receberam peso igual a 1000). Já as restrições fracas recebem pesos menores, que se diferenciam de acordo com as prioridades da instituição. Na implementação feita, que segue o trabalho de De Cescio *et al.* (2010), as restrições *R05* e *R08* receberam peso 1; a restrição *R06* peso 5; e a restrição *R07* recebeu peso igual a 2.

Sejam, então, V o conjunto de restrições do problema com elementos $v \in V$ identificados por $v = 1, \dots, v$; $f_v(X)$ a função que avalia uma solução X com respeito ao atendimento da restrição R_v ; e w_v o peso associado à violação da restrição R_v . A função de avaliação $f_A(X)$ do problema é definida, então, como:

$$f_A(X) = \sum_{v=1}^{v'} [w_v \times f_v(X)] \quad (1)$$

O objetivo do problema é a minimização dessa função.

3.4. Busca Local

Como o espaço de busca do problema é muito amplo, devido à quantidade de combinações e movimentos possíveis, adotou-se, como busca local, o método randômico de descida, descrito em Souza (2000). Trata-se de um método que não pesquisa toda a vizinhança de uma solução (como ocorre no método de descida) e, dessa forma, evita-se a pesquisa exaustiva, que neste problema demandaria um alto tempo de processamento.

O método implementado consiste em analisar um conjunto de vizinhos gerados a partir de movimentos aleatórios, sendo o melhor deles comparado com a solução corrente. A solução vizinha somente será a nova solução corrente caso seja melhor ou igual à anterior. Para geração da vizinhança, a cada iteração todos os movimentos descritos na seção 3.2 têm a mesma probabilidade de ocorrência. O procedimento é repetido por um número fixo de iterações, ressaltando-se que, devido ao vasto espaço de busca, a cada iteração são gerados vizinhos de somente uma disciplina.

A disciplina é escolhida a cada iteração de forma aleatória em uma estrutura de dados do tipo lista. Após ser escolhida, a mesma é removida para que todas as disciplinas passem pelo processo de busca local com a mesma frequência. Quando a lista de disciplinas fica vazia, ela é preenchida novamente com todas as disciplinas em ordem aleatória, até que o

critério de parada seja alcançado. Para esta implementação, foi utilizado, como critério de parada, um valor de número máximo de iterações.

3.5. Geração da Solução Inicial

A solução inicial é gerada pela aplicação da fase de construção da meta-heurística GRASP (FEO e RESENDE, 1995). GRASP é uma meta-heurística proposta por Feo e Resende (1995), que consiste na aplicação iterativa de duas fases: uma fase de construção da solução via um procedimento guloso-aleatório (lista restrita de candidatos e parâmetro α) e outra fase, de busca local, na qual um ótimo local da vizinhança da solução construída é pesquisado. Neste trabalho, fez-se uso apenas da fase de construção do GRASP, a qual é aplicada uma única vez.

O procedimento implementado neste trabalho realiza as alocações de todas as aulas de cada disciplina uma a uma. A alocação é feita a partir de cada disciplina, de modo que todas as aulas de uma disciplina devem primeiramente ser alocadas para somente após realizar as alocações da próxima disciplina. Desta forma, a ordem da lista de disciplinas candidatas (LDC) é de suma importância no procedimento. O ordenamento das disciplinas é determinado por uma função que considera:

- o número de currículos que a disciplina está envolvida. Quanto maior o número de currículos em que uma disciplina está envolvida, maior a chance desta apresentar conflitos;
- o número de indisponibilidades de horários da disciplina;
- a quantidade de alunos na disciplina, uma vez que disciplinas que possuam um alto número de alunos provavelmente terão poucas salas capacitadas.

As alocações são realizadas para todas as disciplinas, respeitando-se o ordenamento da LDC. Para cada disciplina, tem-se o número de aulas a serem alocadas, sendo a alocação feita de forma a testar todas as possíveis salas em todos os possíveis horários. Funções especiais de avaliação (que consideram as mesmas restrições do problema) foram criadas para testar as possíveis alocações. Após isso, uma lista de possíveis alocações candidatas (LPAC) é criada de forma ordenada, de acordo com o valor de avaliação de cada possível alocação.

A partir da LPAC, é criada uma Lista Restrita de Possíveis Alocações Candidatas (LRPAC), sendo o número de elementos dessa última lista definida por um parâmetro α , que varia no intervalo $[0, q]$, sendo q o número máximo de disciplinas a serem alocadas. A LRPAC é composta pelas α melhores possíveis alocações. A seguir, seleciona-se aleatoriamente uma alocação na LRPAC e atribui-se esta alocação à solução que está sendo construída. Como a alocação é escolhida de maneira aleatória, a cada execução uma solução diferente é criada. O procedimento continua até que todas as aulas de todas as disciplinas sejam alocadas.

3.6. ILS

A meta-heurística ILS (*Iterated Local Search*), de acordo com Lourenço *et al.* (2003), é um método de busca iterativa que faz uso de perturbações em ótimos locais, com o objetivo de diversificar e visitar outras regiões do espaço de soluções.

São quatro os principais componentes que definem o método ILS: geração da solução inicial, busca local, perturbação e critério de aceitação. Primeiramente, a geração da solução inicial é realizada por meio da fase de construção da meta-heurística GRASP. Já a busca local é realizada pelo Método de Descida Randômica por Disciplina, conforme descrito na subseção anterior.

Duas formas de se fazer as perturbações em uma solução corrente são utilizadas no

algoritmo implementado. Na primeira, somente o procedimento *Kempeswap* (descrito a seguir na subseção 3.7) é empregado nas perturbações realizadas. Já na segunda forma, os movimentos de geração de vizinhança, juntamente com um novo tipo de movimento, são selecionados aleatoriamente para perturbar uma solução corrente. Níveis de intensidade determinam quantas vezes em uma mesma iteração uma solução corrente será alterada, sendo que o nível um corresponde a 2 alterações, o nível dois a 3 alterações, e assim por diante. Um exemplo de perturbação de nível dois na primeira forma é a aplicação de três chamadas consecutivas ao procedimento *Kempeswap* em uma solução corrente; já na segunda forma, três movimentos escolhidos aleatoriamente são aplicados a uma mesma solução. Deve-se ressaltar que o procedimento *Kempeswap* não faz parte deste tipo de perturbação

O novo movimento criado para a perturbação é denominado *constroiGRASPDisciplina*. Este movimento consiste em desalocar todas as aulas de determinada disciplina e alocá-las novamente, utilizando parte da meta-heurística GRASP construída para criação da solução inicial. Este movimento apresentou grande importância, pois é uma perturbação forte que mantém a qualidade da solução. Contudo, deve-se ressaltar que este movimento apresenta uma probabilidade menor de ocorrência, por apresentar um alto custo computacional.

Quanto ao critério de aceitação, foi utilizada uma condição para aceitar também soluções de piora depois de certo número de soluções sem melhora. Soluções de piora são aceitas quando uma solução corrente permanece inalterada após passar pelos dois níveis de perturbação do ILS sem nenhuma solução que seja menor ou igual à melhor solução corrente.

O controle do percentual de piora para uma solução corrente é feito pela variável denominada *relaxacao*. Esta variável recebe valor zero toda vez que existe uma melhora em relação ao critério de aceitação corrente; caso isso não ocorra em nenhuma das perturbações dos dois níveis do ILS, a variável é incrementada. A variável é utilizada juntamente com *solucao_{best}* no critério de aceitação do ILS, mas seu valor é diferente de nulo somente quando uma solução permanece por algumas iterações sem modificação alguma (na maioria das vezes, quando está presa em uma bacia de atração). Dessa forma, a solução encontrada após a perturbação e busca local passa a ser *solucao_{Best}* somente se for melhor ou igual ao valor da solução corrente.

A utilização deste critério de aceitação é justificada pelo fato de o problema estudado ser caracterizado por vários ótimos locais, os quais dificultam a exploração do espaço de soluções. Busca-se, com esta técnica, aliada às perturbações, uma forma mais eficaz de percorrer o espaço de busca. Testes justificaram a utilização deste critério no ILS, com uma melhora média de 18,6% na qualidade das soluções geradas (testes estes que não foram apresentados neste artigo pela limitação do espaço).

A estrutura, bem como o funcionamento do ILS implementado, pode ser observado e entendido melhor no pseudocódigo apresentado na subseção 3.8.

3.7. O Procedimento *Kempeswap*

De acordo com Lü e Hao (2010), *Kempeswap* é um procedimento definido pela troca de horários entre as aulas de duas cadeias Kempe. Tendo em vista as disciplinas e os conflitos de currículo, cada solução para o problema CBCT pode ser representada por um grafo G no qual os nós são as aulas de cada disciplina e as arestas interligam aulas de um mesmo currículo ou ministradas por um mesmo professor. Uma cadeia Kempe é definida como um conjunto de nós interligados que compõem um subgrafo de G , que é obtido a partir de dois horários específicos.

Formalmente, sendo K_1 e K_2 duas cadeias Kempe no subgrafo gerado a partir dos horários t_i e t_j , o movimento *Kempeswap* produz uma solução substituindo t_i com $(t_i \setminus (K_1 \cup K_2)) \cup (t_j \cap (K_1 \cup K_2))$ e t_j com $(t_j \setminus (K_1 \cup K_2)) \cup (t_i \cap (K_1 \cup K_2))$. Por definição, pelo menos

três disciplinas devem estar envolvidas no movimento, sendo $|K_1|+|K_2| \geq 3$. A Figura 1(a) apresenta um subgrafo gerado a partir de dois períodos t_i e t_j . Neste caso, são encontradas cinco cadeias Kempe: $K_a = \{C_1, C_2, C_7, C_8\}$, $K_b = \{C_3, C_6, C_9\}$, $K_c = \{C_4, C_{11}, C_{12}\}$, $K_d = \{C_5\}$ e $K_e = \{C_{10}\}$. Um exemplo de *Kempeswap* no subgrafo da Figura 1(a) é entre K_b e K_c , produzindo uma nova atribuição movendo $\{C_3, C_4, C_6\}$ para t_j e $\{C_9, C_{11}, C_{12}\}$ para t_i (representado na Figura(b)).

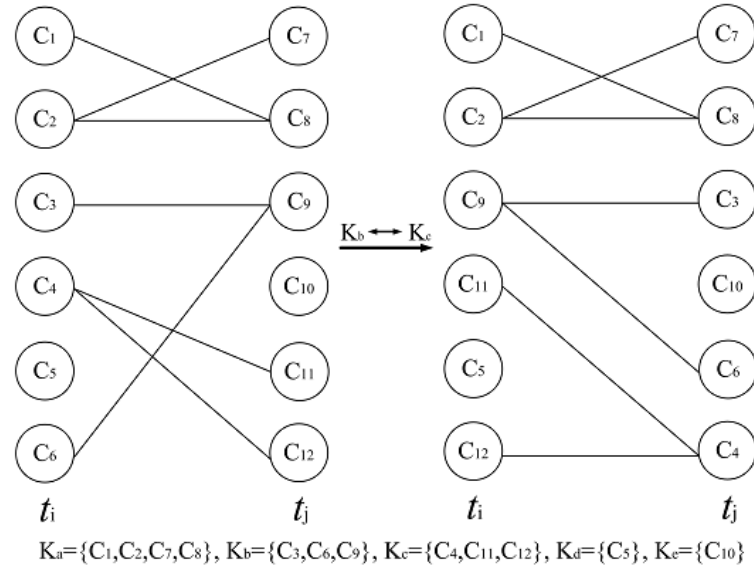


Figura 1: Cadeias Kempe. Fonte: Lü e Hao (2010)

Após a realização do movimento entre as aulas das cadeias Kempe selecionadas, é necessário um procedimento para alocação de salas (que ocorre apenas nas aulas movimentadas). Nos trabalhos relacionados, foram utilizados métodos exatos para esta atribuição. Já na implementação deste trabalho, é utilizada parte da implementação da meta-heurística GRASP para testar todas as possíveis salas a cada aula, realizando alocações que respeitem ao máximo as restrições do problema.

A troca de períodos entre duplas cadeias Kempe é considerada uma generalização do movimento simples Kempe, que consiste de uma troca de períodos envolvendo apenas uma cadeia que apresente mais de três disciplinas. A utilização do movimento simples Kempe pode ser vista em Chiarandini *et al.* (2006) e Thompson e Dowsland (1998). Já a utilização do movimento duplo em cadeias Kempe, denominado *Kempeswap*, é apresentado em Lü e Hao (2010) e Lü *et al.* (2011). Neste segundo trabalho é feita uma comparação entre os movimentos Kempe, juntamente com uma análise dos mesmos, que comprova a eficiência do movimento duplo.

Observa-se que o procedimento *Kempeswap* pode gerar novas soluções modificando várias aulas de diferentes disciplinas, mas mantendo a factibilidade de uma solução. Devido a esta característica, o procedimento foi utilizado como uma forma de perturbação do ILS. Nos trabalhos relacionados, este procedimento é utilizado como um movimento para exploração de sua respectiva vizinhança. Esse tipo de abordagem não foi realizado pelo fato de o procedimento requerer um alto recurso computacional.

3.8. Algoritmo GRASP-ILS

O algoritmo implementado neste trabalho é uma técnica híbrida que utiliza GRASP e ILS. Do GRASP, é utilizada sua fase de construção, para gerar a solução inicial. Sua utilização é de suma importância, pois influencia diretamente os resultados obtidos. Ressalta-se, também, que apenas uma solução é construída como solução inicial. A princípio, várias

soluções eram geradas e aquela com melhor valor de avaliação era escolhida como solução inicial; porém, para algumas instâncias, esta prática demandou um alto tempo computacional, o que inviabilizou sua utilização.

A partir da solução inicial criada, uma busca local é realizada a fim de melhorar ainda mais a solução inicial. Em seguida, o ILS continua a exploração do espaço de soluções por meio da aplicação de perturbações, seguidas de buscas locais. A cada iteração do ILS, o tipo de perturbação é selecionado aleatoriamente e o critério de aceitação pode ou não ser atualizado. O critério de parada é o tempo de processamento. A variável auxiliar S^* guarda a melhor solução encontrada no decorrer de toda a execução da técnica.

O Algoritmo 1 apresenta o pseudocódigo do algoritmo implementado.

Algoritmo 1: GRASP-ILS

```

Entrada:  $\alpha$ , NumVizinhos, MaxIt, TempoDeExecucao, iterPorNivel, numDeNiveis
Saída:  $S^*$ 
1 início
2   relaxacao  $\leftarrow$  0;
3   nivel  $\leftarrow$  1;
4   tempo  $\leftarrow$  0;
5   iteracaoSemMelhora  $\leftarrow$  0;
6   SAux  $\leftarrow$  solução inicia sem nenhuma aula alocada; // Solução Auxiliar
7   SBest  $\leftarrow$  GRASP( $\alpha$ ) // Melhor solução Corrente
8   S*  $\leftarrow$  SBest; // Melhor solução Global
9   enquanto (tempo < TempoDeExecucao) faça
10    SAux  $\leftarrow$  perturbacao(nivel, SBest, tipoAleatorio);
11    SAux  $\leftarrow$  descidaRandomicaPorDisciplina(SAux, NumVizinhos, MaxIt);
12    se SAux  $\leq$  (SBest + relaxacao) então
13      se SAux < (SBest + relaxacao) então
14        relaxacao  $\leftarrow$  0;
15        nivel  $\leftarrow$  1;
16        iteracaoSemMelhora  $\leftarrow$  0
17      fim
18      SBest  $\leftarrow$  SAux;
19      se SBest < S* então
20        S*  $\leftarrow$  SBest;
21      fim
22    senão
23      iteracaoSemMelhora  $\leftarrow$  iteracaoSemMelhora + 1
24      se iteracaoSemMelhora resto iterPorNivel == 0 então
25        nivel  $\leftarrow$  nivel + 1;
26        se nivel > numDeNiveis então
27          nivel  $\leftarrow$  0;
28          relaxacao  $\leftarrow$  relaxacao + 1;
29        fim
30      fim
31    fim
32  fim
33  Retorne S*
34 fim

```

4. RESULTADOS COMPUTACIONAIS

As implementações descritas neste artigo foram feitas sobre a plataforma Java, versão JDK 6.0, utilizando o IDE Netbeans 6.8. Os parâmetros das técnicas foram todos alcançados por meio de testes empíricos. São eles: $\alpha = 2$, NumVizinhos = 15, MaxIt = $5 \times$ \ Número de Disciplinas, iterPorNivel = 4 e numNiveis = 2. Como critério de parada, foi utilizado um tempo máximo de 338 segundos, sendo este tempo calculado de acordo com o hardware em que as instâncias foram executadas. O cálculo é feito por um software fornecido no sítio http://www.cs.qub.ac.uk/itc2007/index_files/benchmarking.htm e faz parte do regulamento do ITC 2007, sendo estas e outras informações descritas no trabalho de Di

Gaspero e Schaerf (2007).

Foram testadas 14 instâncias reais da Universidade de Udine (Itália), as quais são fornecidas pelo site <http://tabu.diegm.uniud.it/ctt/index.php?page=instances>. Os testes foram realizados em um computador com processador Core 2 Quad 2.5 Ghz, memória RAM de 4GB e sistema operacional Windows XP 32 bits. Foram feitas 15 execuções sobre cada instância. Os resultados obtidos, juntamente com a média de valores encontrados e os resultados dos cinco primeiros colocados no ITC 2007 são apresentados na Tabela 2.

Tabela 2: Resultados obtidos com a técnica GRASP-ILS com e sem Kempeswap (resultados riscados indicam valores maiores ou iguais aos alcançados pela técnica com Kempeswap) e média dos resultados encontrados com as técnicas testadas

INSTÂNCIA	Müller	Lu, Hao	Atsuta et. Al	Gelger	Clark et. Al	GRASP ILS	GRASP ILS	GRASP ILS	GRASP ILS
	(USA)	(França)	(Japão)	(Alemanha)	(Singapura)		KEMPESWAP	Média	KEMPESWAP Méd.
<i>comp01.ctt</i>	5	5	5	5	10	5	5	5,00	5,00
<i>comp02.ctt</i>	51	55	50	111	111	94	94	140,67	124,42
<i>comp03.ctt</i>	84	71	82	128	119	112	106	139,58	127,92
<i>comp04.ctt</i>	37	43	35	72	72	43	39	55,00	53,42
<i>comp05.ctt</i>	330	309	312	410	426	405	381	1303,75	978,50
<i>comp06.ctt</i>	48	53	69	100	130	99	94	118,08	114,75
<i>comp07.ctt</i>	20	28	42	57	110	72	67	102,08	95,50
<i>comp08.ctt</i>	41	49	40	77	83	51	49	61,00	59,50
<i>comp09.ctt</i>	109	105	110	150	139	118	113	129,42	124,67
<i>comp10.ctt</i>	16	21	27	71	85	61	58	82,17	77,67
<i>comp11.ctt</i>	0	0	0	0	3	0	0	0,00	0,00
<i>comp12.ctt</i>	333	343	351	442	408	395	381	436,08	434,67
<i>comp13.ctt</i>	66	73	68	98	113	79	83	90,33	93,83
<i>comp14.ctt</i>	59	57	59	90	84	79	70	88,50	82,92
RANK ITC	1	2	3	4	5	-	-		

A utilização do Kempeswap como perturbação do ILS é validada pelos resultados obtidos. Na maioria das instâncias testadas, a implementação que faz uso desse procedimento conseguiu resultados melhores que os da outra técnica testada. Além do mais, a média das soluções encontradas em cada instância mostra que a utilização do procedimento não só produz melhores resultados como também faz com que a técnica seja mais robusta.

Os resultados obtidos também foram satisfatórios frente aos resultados da literatura, tendo em vista que, na maioria das instâncias, os valores alcançados foram melhores que os do 4º e 5º colocados na competição ITC 2007 e, em duas instâncias, conseguiu-se resultados iguais aos do 3º colocado. Deve-se ressaltar, também, que em todas as resoluções são alcançadas soluções factíveis, uma tarefa difícil em algumas instâncias, pela alta probabilidade de conflitos.

Finalmente, todos os resultados obtidos foram validados e cadastrados em <http://tabu.diegm.uniud.it/ctt/> de acordo com as normas do ITC 2007.

5. CONCLUSÕES E TRABALHOS FUTUROS

Este artigo apresenta a técnica GRASP-ILS com utilização do procedimento *Kempeswap* para resolução do Problema de Programação de Cursos Universitários baseadas em Currículos. A técnica implementada foi testada e comparada com uma implementação anterior, que não faz uso desse procedimento, e também com outros resultados da literatura, sendo dessa forma avaliada a eficiência da abordagem realizada. Os resultados obtidos foram

satisfatórios, tendo em vista as comparações realizadas.

Os principais pontos que contribuíram para eficiência da abordagem realizada foram: a utilização do GRASP para geração de uma solução inicial de qualidade; o procedimento de descida aleatória, que realiza uma busca local eficiente e rápida; a utilização de movimentos inteligentes que desconsideram um bom número de vizinhos ineficazes; as perturbações implementadas e também o critério de aceitação implementado no ILS, o que faz com que a meta-heurística se comporte de maneira adaptativa, aceitando soluções de piora em determinadas situações.

O procedimento *Kempeswap* pode ser melhor explorado, uma vez que se trata de um movimento avançado que pode percorrer espaços de soluções de maneira eficiente, escapando de alguns ótimos locais que os demais movimentos não conseguiriam escapar. Porém, é um procedimento custoso computacionalmente que, por isso, deve ter seu uso otimizado, para que possa ser usado com maior frequência e eficiência. Dessa forma, a otimização e o emprego do procedimento em questão em novas atribuições, juntamente com a implementação de novas meta-heurísticas, como Busca Tabu, podem ser consideradas etapas futuras deste trabalho.

Por fim, ressalta-se a importância deste trabalho, uma vez que os resultados obtidos e a metodologia apresentada servem de base para novos estudos, tendo em vista que o procedimento *Kempeswap* vem sendo utilizado na literatura com certa eficiência para diferentes problemas. Os resultados estão disponíveis, incluindo os apresentados no presente artigo, no endereço: <http://tabu.diegm.uniud.it/ctt/>.

AGRADECIMENTOS

Os autores agradecem ao CEFET-MG, FAPEMIG, CAPES e CNPq, pelo apoio recebido.

REFERÊNCIAS

BAI, R.; BURKE, E. K.; KENDALL, G. e MCCULLUM, B. (2008). A simulated annealing hyper-heuristic for university course timetabling problem extended abstract. (Abstract) **PATAT'06, Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling**, p. 345–350.

BARBOSA, S. H. D.; SOUZA, S. R. e SOUSA, A. M. (2010). Utilização de uma heurística híbrida para resolução do problema de alocação de horários em uma universidade. **XIII Encontro de Modelagem Computacional (EMC 2010)**. Nova Friburgo - RJ.

BURKE, E. K. e PETROVIC, S. (2002). Recent research directions in automated timetabling. **European Journal of Operational Research**, v. 140, p. 266–280.

CHIARANDINI, M.; BIRATTARI, M.; SOCHA, K. e ROSSI-DORIA, O. (2006). An effective hybrid algorithm for university course timetabling. **J. of Scheduling**, v. 9, p. 403–432.

DE CESCO, F.; DI GASPERO, L. e SCHAERF, A. (2010). Benchmarking curriculum-based course timetabling: formulations, data formats, instances, validation, visualization, and results. **Annals of Operations Research**, v. 173, n. 1, p. 1–12.

DI GASPERO, L. e SCHAERF, A. (2007). **The second international timetabling**

competition (ITC 2007): Curriculum-based course timetabling (track 3). Technical Report QUB/IEEE/Tech/ITC2007/CurriculumCTT/v1.0/1, School of Electronics, Electrical Engineering and Computer Science, Queens University, Belfast (UK). ITC-2007 site: <http://www.cs.qub.ac.uk/itc2007/>.

FEO, T. A. e RESENDE, M. G. C. (1995). Greedy randomized adaptive search procedures. **Journal of Global Optimization**, v. 6, p. 109–133.

LÜ, Z. e HAO, J. (2010). Adaptive tabu search for course timetabling. **European Journal of Operational Research**, v. 200, n. 1, p. 235 – 244.

LÜ, Z.; HAO, J. e Glover, F. (2011). Neighborhood analysis: a case study on curriculum-based course timetabling. **Journal of Heuristics**, v. 17, p. 97–118.

LEWIS, R. (2008). A survey of metaheuristic-based techniques for university timetabling problems. **OR Spectrum**, v. 30, p. 167–190.

LOURENÇO, H. R.; MARTIN, O. e STÜTZLE, T. (2003). Iterated Local Search. Glover, F. e Kochenberger, G., editors, **Handbook of Metaheuristics**, p. 321–353. Kluwer Academic Publishers.

MULLER, T. (2008). Itc 2007 solver description: A hybrid approach. **In Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling, PATAT**.

QU, R. (2002). **Case Based Reasoning for Course Timetabling Problems**. PhD thesis, University of Nottingham.

SCHAERF, A. (1999). A survey of automated timetabling. **Artif. Intell. Rev**, v. 13, n. 2, p. 87–127.

SOUZA, M. J. F. (2000). **Programação de horários em escolas: uma aproximação por metaheurísticas**. Tese de doutorado, Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ, Rio de Janeiro.

THOMPSON, J. M. e DOWSLAND, K. A. (1998). A robust simulated annealing based examination timetabling system. **Comput. Oper. Res.**, v. 25, p. 637–648.

WILLEMEN, R. J. (2002). **School timetable construction algorithms and complexity**. Tese de Mestrado, Technische Universiteit Eindhoven.