

Algoritmo Grasp para o Problema de Tabela-horário de Universidades

Wallace S. Rocha

Universidade Federal do Espírito Santo
Vitória – ES – Brasil
walacesrocha@yahoo.com.br

Maria C. S. Boeres

Departamento de Informática – Universidade Federal do Espírito Santo
Vitória – ES – Brasil
boeres@inf.ufes.br

Maria C. Rangel

Departamento de Informática – Universidade Federal do Espírito Santo
Vitória – ES – Brasil
crangel@inf.ufes.br

RESUMO

Este modelo resume as normas de formatação para os trabalhos completos a serem publicados nos Anais do XLI SBPO. O Resumo deve ter no máximo 150 palavras. Título, nomes dos autores, sua filiação e endereços, resumo e palavras-chave devem repetir fielmente o que foi informado quando o artigo foi cadastrado através do sistema JEMS. No primeiro upload, por ocasião da submissão do trabalho, exclua desta primeira página os nomes, instituições e endereços dos autores.

PALAVRAS CHAVE. Tabela-horário, Meta-heurística, Grasp, Área de classificação principal (escolher uma na janela de áreas).

ABSTRACT

This model summarizes the formatting rules for full papers accepted for inclusion in the Proceedings of the XLI SBPO. The Abstract must have no more than 150 words. Title, author names, affiliations, addresses, abstract and keywords must precisely mirror the information provided during registration. When you initially submit the paper, suppress author names, affiliations and addresses.

KEYWORDS. Timetabling, Metaheuristic, Grasp, Main area (choose between those in the areas window).

1. Introdução

Um dos problemas de grande interesse na área de otimização combinatória é o problema de tabela-horário. Dado um conjunto de disciplinas, alunos, professores e espaço físico, o problema consiste basicamente em alocar os horários das disciplinas levando-se em conta o espaço e horários disponíveis, além de respeitar algumas restrições. Essas restrições surgem das particularidades de cada instituição. Elas podem ser fortes ou fracas. As fortes são aquelas que sempre devem ser satisfeitas, como por exemplo, exigir que um aluno não pode assistir mais de uma aula no mesmo horário. Uma tabela-horário é dita viável quando não viola nenhuma restrição forte. As fracas são restrições que não geram inviabilidade, mas refletem algumas preferências dos professores, alunos ou até mesmo da escola, por exemplo, penalizando uma tabela-horário que tenha aulas isoladas. Quanto mais restrições fracas forem satisfeitas, melhor será a tabela-horário. A formulação usada neste trabalho é a adotada pela competição ITC-2007.

A meta-heurística GRASP (Greedy Randomized Adaptive Search Procedure) é uma técnica de destaque no ramo de otimização combinatória. Ela tem sido aplicada em problemas de cobertura de conjuntos, árvore geradora, entre outros. Contudo, há pouca pesquisa desse algoritmo nos problemas de tabela-horário. Foram encontradas apenas implementações para a modelagem de escolas, mas nada para a de universidades. Este trabalho implementa um algoritmo GRASP para geração de tabela-horário usando a formulação do ITC-2007. O algoritmo possui uma fase inicial onde é garantido que uma tabela-horário viável será encontrada. Na segunda fase ocorre o melhoramento da tabela-horário através de busca local. Uma terceira fase de intensificação ocorre usando o procedimento *path-relinking*. Esse procedimento de três fases se repete algumas iterações, sempre gerando uma nova tabela. A melhor de todas encontradas é a resposta final. Os resultados obtidos são comparados com as melhores soluções publicadas na literatura para este problema.

2. O Problema de Tabela-horário para Universidades (THU)

Apesar das formulações encontradas na literatura para o problema de tabela-horário (THU) serem bastante variadas, elas podem ser agrupadas em três grupos: tabela-horário de escolas, tabela-horário de universidades e tabela-horário para aplicação de exames. Neste trabalho é desenvolvido um algoritmo para tratar a alocação de tabela-horário de universidades. A formulação do problema é a mesma adotada na competição ITC-2007. A principal vantagem em se adotar esta formulação é que muitos autores convergiram para ela, o que contribuiu grandemente para comparação de resultados de diversos pesquisadores.

O problema de tabela-horário nesta formulação consiste das seguintes entidades:

- **Dias, Horários e Períodos:** É dado o número de dias na semana em que há aula (geralmente 5 ou 6). Cada dia é dividido em um número fixo de horários que é igual para todos os dias. Um período é um par composto de um dia e um horário. O total de períodos é obtido multiplicando a quantidade de dias pela quantidade de horários do dia.
- **Disciplinas e Professores:** Cada disciplina possui uma quantidade de aulas semanais que devem ser alocadas em períodos diferentes. É lecionado por um professor e assistida por um dado número de alunos. Cada disciplina deve ser dada em um número mínimo de dias, ou seja, não podem ser lecionadas em poucos dias. Cada disciplina pode ser indisponíveis em alguns períodos.

- Salas: Cada sala possui uma capacidade diferente de assentos.
- Currículo: Um currículo é um grupo de disciplinas que possuem alunos em comum.

A solução do problema é alocação das aulas das disciplinas a um período e uma sala. A solução deve atender a algumas restrições separadas em dois grupos: fortes e fracas.

2.1. Restrições Fortes

As restrições fortes são as seguintes:

- Aulas das Disciplinas: Todas as aulas das disciplinas deve ser alocadas e em períodos diferentes. Uma violação ocorre se uma aula não é alocada ou se duas são alocadas no mesmo período.
- Conflitos: Aulas de disciplinas do mesmo currículo ou lecionadas pelo mesmo professor devem ser alocadas em períodos diferentes.
- Ocupação de Sala: Duas aulas não podem ser dadas na mesma sala e no mesmo período.
- Disponibilidade de Professor: Se um professor é indisponível em um horário, nenhuma disciplina que ele leciona deve ser alocada naquele horário.

2.2. Restrições Fracas

As restrições fracas são as seguintes:

- Capacidade da Sala: O número de alunos da disciplina deve ser menor ou igual ao número de assentos da sala em que a aula for dada. Cada aluno excedente contabiliza uma violação.
- Dias Mínimos de Trabalho: As aulas de cada disciplina devem ser espalhadas por uma quantidade mínima de dias. Cada dia abaixo do mínimo é contado como uma violação.
- Aulas Isoladas: Aulas do mesmo currículo devem ser alocadas em períodos adjacentes. Cada aula isolada é contada como uma violação.
- Estabilidade de Sala: Todas as aulas de uma disciplina devem ser dadas na mesma sala. Cada sala usada para aula que é diferente da primeira é contada como uma violação.

Na contagem total das violações fracas são considerados pesos diferentes para cada tipo de violação. A restrição de dias mínimos possui peso cinco, aulas isoladas peso dois e as demais peso um.

3. Trabalhos Relacionados

As principais técnicas de solução do problema THU recaem basicamente em três grupos: programação matemática, programação em lógica e meta-heurísticas. Por ser um problema complexo, é inviável computacionalmente buscar a solução ótima. Alguns métodos tem alcançado este objetivo, mas apenas para instâncias pequenas. Como pode ser visto em [Schaerf (1995)], este problema é NP-completo, portanto não existe um algoritmo de complexidade polinomial para encontrar a solução ótima.

Na área de programação matemática, bons resultados têm sido obtidos com programação inteira [Lach (2010)], [van den Broek (2010)] e [Burke (2008)].

Algumas formulações do problema THU usando programação em lógica podem ser vistas em [Gueret (1995)], [Goltz (1999)] e [Asín Achá (2010)].

A maioria das técnicas de solução deste problema utilizam meta-heurísticas. Ainda não foi identificada uma meta-heurística que seja considerada a melhor. Algoritmos eficientes e que produzem bons resultados têm sido encontrados usando Simulated Annealing [Kostuch (2006)], [Bai (2012)], [Elmohamed (1998)], Algoritmo Genético [Erben (1995)], [Suyanto (2010)], [Kano (2008)], Busca Tabu [Elloumi (2008)], entre outras mais recentes como a Busca de Harmonia [Al-Betar (2012)].

Há também algumas técnicas híbridas em que são usadas mais de uma meta-heurística ou elas são aplicadas de forma diferente. Exemplos deste tipo de implementação podem ser visto em [Massoodian (2008)] em que o algoritmo genético é combinado com um algoritmo de busca local para melhorar a qualidade das soluções. Em [Kostuch (2006)] há um algoritmo que constói a tabela-horário em três etapas, cada uma executando um procedimento de Simulated Annealing independente.

Este trabalho propõe um algoritmo GRASP para o problema de tabela-horário de universidades. Há na literatura algumas propostas do GRASP, mas para tabela-horário de escolas [Souza (2004)], [Vieira (2010)].

4. Algoritmos para o THU

4.1. Algoritmo Genético

4.2. Simulated Annealing

4.3. GRASP - com a primeira busca local

As disciplinas são ordenadas por ordem de dificuldade. Da mais difícil para a mais fácil elas são selecionadas uma a uma para entrar na tabela. Para escolher o horário e a sala da disciplina são avaliadas todas as opções viáveis e é feita uma contagem para verificar quantas violações fracas haverá com cada escolha. A partir desta avaliação é montada a lista restrita de candidatos, de onde será escolhida a posição de inserir a disciplina. Se ao tentar inserir uma disciplina na tabela não houver uma posição viável, ocorre uma “explosão” da tabela. É uma estratégia que retira da tabela uma aula alocada anteriormente para abrir espaço para a disciplina que não está sendo possível alocar. Com essa estratégia garante-se que ao final da primeira fase uma tabela-horário viável será encontrada.

5. Resultados Computacionais

6. Conclusões e Trabalhos Futuros

O SBPO utilizará este ano, novamente, o sistema JEMS para a submissão de trabalhos. A primeira página dos trabalhos publicados será constituída com as informações fornecidas no formulário de submissão de trabalho do JEMS. Por isto, os nomes de **todos** os autores devem ser cadastrados nesse formulário. Os nomes não incluídos nesse formulário não aparecerão entre os autores na programação do Simpósio nem nos Anais. No campo *Paper Title* deve ser informado apenas o título do trabalho, **sem qualquer identificação dos autores ou suas instituições**. O campo *Paper Abstract* deverá ser preenchido com o Resumo, de **no máximo 150 palavras**, seguido por 3 palavras-chave e pelo nome da área de classificação principal do trabalho, escolhida entre aquelas assinaladas no campo *Paper Topics*. A seguir, no caso de resumos escritos em português ou espanhol, deverá vir o *Abstract* em inglês e as *keywords*, traduzindo fielmente o Resumo e as palavras-chave.

7. Submissão do Texto Completo

Após cadastrar o artigo, o autor é convidado a carregar para o sistema JEMS arquivo de terminação DOC ou PDF, com o texto completo. A primeira página desse manuscrito deve conter apenas o título do artigo coincidindo exatamente com o informado quando do cadastramento. Pode, também, incluir novamente os resumos e palavras-chave em português ou espanhol e inglês, mas, **não pode incluir nomes, filiação e endereços de autores**. Este manuscrito será disponibilizado para o exame pelos revisores, que terão também acesso às informações do cadastro, exceto as referentes aos nomes e instituições dos autores. Uma vez aceito o artigo, os autores serão chamados a encaminhar **versão final** com a página inicial completa, isto é, com autores, instituições, resumo de no máximo 150 palavras, 3 palavras-chave, *abstract* e *keywords*. As páginas deste texto não devem vir numeradas, tanto no caso de arquivo enviado quando da submissão quanto no caso do arquivo com a versão final do artigo aceito. A numeração será feita posteriormente para o conjunto de todos os artigos. **Cabeçalhos e rodapés devem ser deixados em branco.**

8. Instruções de Formatação

Os trabalhos completos devem ter, **no máximo 12 páginas**, tudo incluído neste limite, inclusive tabelas, gráficos, agradecimentos e referências. Os textos devem utilizar páginas de tamanho **A4** (29,7 x 21,0 cm) **com margem superior de 3,3 cm, inferior de 2,5 cm e laterais de 2,9 cm**. Devem ser escritos em coluna única, com fonte **Times 11**.

9. Estilo das Citações

As citações no texto devem conter o último sobrenome do autor, seguido, entre parêntesis, do **ano da publicação**, como por exemplo [Knuth (1984)], [Boulic (1991)], e [Smith (1999)]. As referências no final do texto devem estar em ordem alfabética do último sobrenome do primeiro autor.

Referências

- Al-Betar, M. and Khader, A. (2012). A harmony search algorithm for university course timetabling. *Annals of Operations Research*, 194:3–31. 10.1007/s10479-010-0769-z.
- Asín Achá, R. and Nieuwenhuis, R. (2010). Curriculum-based course timetabling with sat and maxsat. *Annals of Operations Research*, pages 1–21. 10.1007/s10479-012-1081-x.
- Bai, R., Blazewicz, J., Burke, E., Kendall, G., and McCollum, B. (2012). A simulated annealing hyper-heuristic methodology for flexible decision support. *4OR: A Quarterly Journal of Operations Research*, 10:43–66. 10.1007/s10288-011-0182-8.
- Boulic, R. and Renault, O. (1991). 3d hierarchies for animation. In Magnenat-Thalmann, N. and Thalmann, D., editors, *New Trends in Animation and Visualization*. John Wiley & Sons Ltd.
- Burke, E. K., Mareček, J., Parkes, A. J., and Rudová, H. (2008). A branch-andcut procedure for the udine course timetabling. In *Problem, Proceedings of the 7th PATAT Conference, 2008*.
- Elloumi, A., Kamoun, H., Ferland, J., and Dammak, A. (2008). A tabu search procedure for course timetabling at a tunisian university. In *Proceedings of the 7th PATAT Conference, 2008*.

- Elmohamed, M. A. S., Coddington, P., and Fox, G. (1998). A comparison of annealing techniques for academic course scheduling. In *Lecture Notes in Computer Science*.
- Erben, W. and Keppler, J. (1995). A genetic algorithm solving a weekly course-timetabling problem.
- Goltz, H.-J. and Matzke, D. (1999). University timetabling using constraint logic programming. In *PRACTICAL ASPECTS OF DECLARATIVE LANGUAGES*, pages 320–334. Springer-Verlag.
- Gueret, C., Jussien, N., Boizumault, P., and Prins, C. (1995). Building university timetables using constraint logic programming.
- Kanoh, H. and Sakamoto, Y. (2008). Knowledge-based genetic algorithm for university course timetabling problems. *Int. J. Know.-Based Intell. Eng. Syst.*, 12(4):283–294.
- Knuth, D. E. (1984). *The T_EX Book*. Addison-Wesley, 15th edition.
- Kostuch, P. (2006). The university course timetabling problem with a 3-phase approach.
- Lach, G. and Lubbecke, M. E. (2010). Curriculum based course timetabling: new solutions to udine benchmark instances. *Annals of Operations Research*.
- Massoodian, S. and Esteki, A. (2008). A hybrid genetic algorithm for curriculum based course timetabling. In *Proceedings of the 7th PATAT Conference, 2008*.
- Schaerf, A. (1995). A survey of automated timetabling. *ARTIFICIAL INTELLIGENCE REVIEW*, 13:87–127.
- Smith, A. and Jones, B. (1999). On the complexity of computing. In Smith-Jones, A. B., editor, *Advances in Computer Science*, pages 555–566. Publishing Press.
- Souza, M. J. F., Maculan, N., and Ochi, L. S. (2004). Metaheuristics. chapter A GRASP-tabu search algorithm for solving school timetabling problems, pages 659–672. Kluwer Academic Publishers, Norwell, MA, USA.
- Suyanto (2010). An informed genetic algorithm for university course and student timetabling problems. In Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L., and Zurada, J., editors, *Artificial Intelligence and Soft Computing*, volume 6114 of *Lecture Notes in Computer Science*, pages 229–236. Springer Berlin / Heidelberg.
- van den Broek, J. and Hurkens, C. (2010). An ip-based heuristic for the post enrolment course timetabling problem of the itc2007. *Annals of Operations Research*.
- Vieira, A., Rafael, M., and Scaraficci, A. (2010). A grasp strategy for a more constrained school timetabling problem.