

Walace de Souza Rocha

***Algoritmo GRASP para o Problema de
Tabela-horário de Universidades***

Vitória - ES, Brasil

28 de Fevereiro de 2013

Walace de Souza Rocha

***Algoritmo GRASP para o Problema de
Tabela-horário de Universidades***

Dissertação apresentada para obtenção do Grau
de Bacharel em Ciência da Computação pela
Universidade Federal do Espírito Santo.

Orientador:

Maria Claudia Silva Boeres

Co-orientador:

Maria Cristina Rangel

DEPARTAMENTO DE INFORMÁTICA
CENTRO TECNOLÓGICO
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Vitória - ES, Brasil

28 de Fevereiro de 2013

Dissertação de Projeto Final de Graduação sob o título “*Algoritmo GRASP para o Problema de Tabela-horário de Universidades*”, defendida por Wallace de Souza Rocha e aprovada em 28 de Fevereiro de 2013, em Vitória, Estado do Espírito Santo, pela banca examinadora constituída pelos professores:

Profa. Dr. Maria Claudia Silva Boeres
Orientadora

Profa. Dr. Maria Cristina Rangel
Co-orientadora

Prof. Dr. Fulano de Tal
Universidade Federal do Espírito Santo

Resumo

Escreva aqui o texto do seu resumo.

Abstract

Write here the English version of your “Resumo”.

Dedicatória

Dedico este trabalho a ...

Agradecimentos

Agradeço a ...

Sumário

Lista de Figuras

Lista de Tabelas

| | | |
|----------|---|-------|
| 1 | Introdução | p. 11 |
| 1.1 | Apresentação do problema | p. 12 |
| 1.2 | Motivação para o problema | p. 12 |
| 1.3 | Objetivos deste trabalho | p. 12 |
| 1.4 | Organização do texto | p. 13 |
| 2 | Estado da Arte | p. 14 |
| 2.1 | Introdução | p. 15 |
| 2.2 | Conceitos e definições | p. 15 |
| 2.3 | Abordagens existentes | p. 15 |
| 2.4 | Trabalhos relacionados | p. 17 |
| 3 | Proposta do trabalho | p. 18 |
| 3.1 | Formulação do ITC-2007 | p. 19 |
| 3.2 | Algoritmo GRASP | p. 19 |
| 4 | Resultados Computacionais | p. 20 |
| 4.1 | Descrição das instâncias utilizadas | p. 21 |
| 4.2 | Detalhes de implementação | p. 21 |
| 4.3 | Sintonia da Meta-heurística | p. 21 |

| | | |
|----------|---|-------|
| 4.4 | Análise dos resultados | p. 22 |
| 5 | Conclusões e trabalhos futuros | p. 23 |
| 5.1 | Conclusões | p. 24 |
| | Referências Bibliográficas | p. 25 |
| | Anexo A – Ferramentas utilizadas | p. 27 |

Lista de Figuras

Lista de Tabelas

| | | |
|-----|---|------|
| 4.1 | Tabela com informações sobre cada instância do ITC-2007 | p.22 |
|-----|---|------|

1 Introdução

Neste capítulo são apresentados o objetivo desta dissertação e a estrutura da mesma.

1.1 Apresentação do problema

O problema de tabela-horário consiste em alocar um conjunto de aulas em um número pré-determinado de horários, satisfazendo diversas restrições envolvendo professores, alunos e o espaço físico disponível. A solução manual deste problema não é uma tarefa trivial e as instituições de ensino precisam resolvê-lo anualmente ou semestralmente. Nem sempre a alocação manual é satisfatória, por exemplo, quando um aluno não consegue matricular em duas disciplinas porque elas são alocadas no mesmo horário.

Por esta razão, atenção especial tem sido dada a solução automática de tabela-horário. Nos últimos cinquenta anos, começando com [Gotlieb 1962], este problema ganhou grande destaque na área de otimização combinatória, tendo diversos trabalhos publicados.

classificação

1.2 Motivação para o problema

O problema de tabela-horário está entre os mais difíceis da área de otimização combinatória. Em [Schaerf 1995] pode ser visto que ele é classificado como NP-completo. Assim, a solução exata só pode ser garantida para instâncias bem pequenas, que não correspondem às instâncias reais da maioria das instituições de ensino.

Existe uma necessidade de propor algoritmos cada vez mais eficientes que produzam tabelas-horário satisfatórias em um tempo viável, independente do tamanho da instância. Devido à complexidade do problema, métodos exaustivos são descartados. Diferentes meta-heurísticas tem sido aplicadas devido ao fato de serem relativamente simples de implementar e produzirem bons resultados. Dentre as meta-heurísticas que já foram aplicadas ao problema, existe um esforço em aplicar melhorias para conseguir melhores resultados. Há algumas meta-heurísticas que ainda não foram exploradas no problema, o que deixa uma incógnita quanto à sua eficiência.

1.3 Objetivos deste trabalho

O objetivo principal deste trabalho é resolver o problema de tabela-horário de universidades usando a meta-heurística GRASP (*Greedy Randomized Adaptive Search Procedures*). Pelo fato de existirem diversas formulações para o problema, será escolhida uma que tem sido bastante utilizada na área: a que é proposta no ITC-2007 (*International Timetabling Competition - 2007*) [PATAT 2008]. A razão principal desta escolha é facilitar a comparação dos resultados com

outros algoritmos propostos na literatura.

Para atingir este objetivo será necessário estudar a formulação do ITC-2007 e propor uma implementação eficiente dos dados. Além de aplicar o GRASP para o problema, pretende-se implementar algumas melhorias que já foram propostas na literatura e que visam melhorar a versão básica do algoritmo.

Deseja-se por fim coletar na literatura resultados obtidos para o problema com diferentes técnicas de solução, a fim de comparar com o algoritmo proposto.

1.4 Organização do texto

No capítulo 2 é apresentado o estado da arte listando as principais técnicas de solução do problema. No capítulo 3 são apresentados a formulação do problema segundo o ITC-2007, o algoritmo GRASP e sua aplicação no problema. No capítulo 4 são apresentados detalhes de implementação, instâncias de testes e os resultados obtidos. No capítulo 5 são listadas as conclusões obtidas no trabalho e enumerados alguns trabalhos futuros.

2 *Estado da Arte*

“Navegar é preciso, viver não.”

Luís de Camões

Neste capítulo são apresentados

2.1 Introdução

O problema de tabela-horário não possui uma formulação única. Como pode ser visto em [Schaerf 1995], ao longo do tempo surgiram diversas modelagens. Essa variedade surgiu pelo fato das restrições do problema serem específicas a determinada instituição de ensino. Alguns *benchmarks* foram criados, como é o caso do ITC-2007 [PATAT 2008]. A maior contribuição do *benchmark* é fornecer um conjunto de dados para que os diferentes pesquisadores possam comparar suas técnicas de solução.

Apesar das diferentes formulações, os problemas de tabela-horário possuem uma característica em comum: a separação das restrições em dois grupos. São fortes ou fracas.

As restrições fortes são aquelas que não podem ser violadas. Elas restringem o conjunto de soluções para impedir certas situações irreais, como por exemplo, alunos assistindo mais de uma aula no mesmo tempo ou uma sala sendo alocada para mais de uma aula no mesmo horário. Se uma tabela-horário não viola nenhuma restrição forte ela é dita ser uma solução viável.

As restrições fracas são aquelas que não interferem na viabilidade da solução, mas refletem certas preferências das instituições. É desejável, por exemplo, que um aluno não tenha aulas isoladas durante o dia ou que aulas da mesma disciplina sejam lecionadas na mesma sala. As restrições fracas podem possuir pesos diferentes.

O objetivo do problema é encontrar uma tabela-horário viável e que minimize a quantidade de violações fracas, portanto, um problema de minimização.

2.2 Conceitos e definições

2.3 Abordagens existentes

As técnicas usadas para resolver o problema de tabela-horário são bastante diversificadas. As mais antigas, as heurísticas construtivas [Junginger 1986, Papoulias 1980], foram baseadas no modo humano de resolver o problema: as aulas eram alocadas uma a uma até finalizar a construção da tabela. Conflitos eram resolvidos realizando trocas de aulas.

[Neufeld e Tartar 1974] propôs uma redução do problema ao problema de coloração de grafos. As aulas são representadas como vértices do grafo. Arestas conectam aulas que não podem ser alocadas no mesmo horário. A coloração do grafo resultante é então transformada na tabela-horário: cada cor representa um horário de aula. Propostas similares a esta foram

aplicadas em [Werra 1985, Selim 1988].

[Ostermann e Werra 1982] reduziram o problema de tabela-horário ao problema de fluxo de redes. As aulas são representadas pelos vértices. Uma rede é criada para cada horário e o fluxo na rede identifica as aulas que são lecionadas no mesmo horário. Posteriormente, [Werra 1985] usou uma idéia similar, mas cada fluxo representa uma classe e os vértices são horários e professores. Outras abordagens com fluxo de redes foram usadas em [Dinkel, Mote e Venkataramanan 1989, Chahal e Werra 1989].

De uns anos para cá as técnicas de solução para o problema têm recaído basicamente em três grupos: programação matemática, programação em lógica e principalmente meta-heurísticas.

Na área de programação matemática, bons resultados têm sido obtidos com programação inteira. Em [Lach e Lubbecke 2010] pode ser vista uma formulação completa do problema. Ela foi executada com CPLEX9 [IBM 2012] e conseguiu soluções com respostas bem próximas às melhores obtidas na competição ITC-2007. Em [Burke et al. 2008] pode ser vista outra formulação com algumas relaxações. Com um tempo de execução aproximado de quinze minutos no CPLEX10, esse algoritmo conseguiu encontrar a solução ótima para duas instâncias da competição. Uma grande contribuição destes dois trabalhos é que eles forneceram limites inferiores para a quantidade de violações das restrições fracas para cada instância do ITC-2007. Em [Filho e Lorena 2006] pode ser vista uma modelagem em programação inteira que trata algumas particularidades das escolas brasileiras de ensino fundamental.

Alguns resultados relevantes também tem sido encontrados com programação em lógica. [Achá e Nieuwenhuis 2010] apresenta uma formulação usando *MaxSAT* em que se conseguiu melhorar quase metade das respostas que eram conhecidas à época para as instâncias do ITC-2007. [Gueret et al. 1995] e [Goltz e Matzke 1999] adotam formulações diferentes do ITC-2007, mas destacam como programação em lógica combinada com programação por restrições podem implementar modelos bem flexíveis, em que restrições podem ser adicionadas, modificadas ou excluídas com pouca alteração de código-fonte.

No ramo da inteligência artificial as meta-heurísticas têm sido bastante utilizadas, abrangendo grande parte dos trabalhos recentes na área. Grande parte do sucesso das meta-heurísticas se deve à estrutura simples do algoritmo e também pelos bons resultados alcançados. *Simulated Annealing*, Algoritmo Genético e Busca Tabu são as mais aplicadas. Há propostas com meta-heurísticas menos conhecidas, como a Busca de Harmonia.

Algoritmos eficientes que produzem bons resultados têm sido encontrados usando *Simulated Annealing* [?], [?], [?], Algoritmo Genético [?], [?], [?], Busca Tabu [?], entre outras mais

recentes como a Busca de Harmonia [?].

Há também algumas técnicas híbridas em que são usadas mais de uma meta-heurística ou aplicadas de forma diferente. Exemplos deste tipo de implementação podem ser visto em [?] em que o algoritmo genético é combinado com um algoritmo de busca local para melhorar a qualidade das soluções. Em [?] há um algoritmo que constrói 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 [?] e [?]. Além disso propõe melhorias para um algoritmo genético e um *Simulated Annealing* encontrados na literatura e aplicados à mesma formulação do problema.

2.4 Trabalhos relacionados

3 *Proposta do trabalho*

“Nada se cria, nada se perde, tudo se transforma.”

Lavousier

Neste capítulo é apresentado a proposta do trabalho ...

3.1 Formulação do ITC-2007

3.2 Algoritmo GRASP

4 Resultados Computacionais

“Nada se cria, nada se perde, tudo se transforma.”

Lavousier

Neste capítulo é apresentado os resultados computacionais ...

4.1 Descrição das instâncias utilizadas

As instâncias utilizadas foram as mesmas submetidas aos competidores do ITC-2007. São 21 instâncias ao todo, com grau de dificuldade variado. A organização garante que existe solução viável para todas as instâncias, fato que foi comprovado nos testes. Mas nada foi informado sobre a quantidade de violações fracas em cada instância. Em [PATAT 2008] podem ser obtidas todas as instâncias.

Na tabela 4.1 são apresentados os dados mais relevantes de cada instância. A quantidade de horários de aula numa semana não varia muito de instância para instância. A coluna conflitos conta a quantidade de pares de aula que não podem ser alocadas no mesmo horário (mesma disciplina, mesmo currículo ou mesmo professor) dividido pelo total de pares distintos de aula. A disponibilidade mede percentualmente a quantidade de horários que são disponíveis para as aulas, levando-se em conta as restrições de indisponibilidade que são informadas no arquivo de entrada.

A quantidade de currículos e disciplinas tem grande impacto no tempo de execução do algoritmo, pois são mais aulas para fazer a contagem total de violações. A quantidade de conflitos e disponibilidade influencia na dificuldade de encontrar uma solução viável, pois quanto mais conflitos e menos disponibilidade, menos horários existirão para alocar aula sem violar as restrições fortes. Além de dificultar a viabilidade no momento de geração da solução inicial, os conflitos e as disponibilidades dificultam a exploração da vizinhança na busca local, dado que muitas trocas acabam sendo descartadas por introduzirem violações das restrições fortes.

4.2 Detalhes de implementação

4.3 Sintonia da Meta-heurística

Duas estruturas de vizinhança foram usadas neste trabalho para a fase de busca local: *MOVE* e *SWAP*. Em [Ceschia, Gaspero e Schaerf 2011] essas duas estruturas também são usadas, só que aplicadas de maneira diferente. Na geração do vizinho sempre é usado o *MOVE*. Em alguns casos ocorre *SWAP* juntamente com o *MOVE*. Essa probabilidade de ocorrer o *SWAP* foi parametrizada. Experimentalmente foi escolhido um valor de 40%.

Testes mostraram que aplicar *MOVE* e *SWAP* separadamente é mais eficiente. Foi obser-

| Instância | Currículos | Salas | Disciplinas | Horários por dia | Dias | Conflitos | Disponibilidade |
|------------------|-------------------|--------------|--------------------|-------------------------|-------------|------------------|------------------------|
| comp01 | 14 | 6 | 30 | 6 | 5 | 13.2 | 93.1 |
| comp02 | 70 | 16 | 82 | 5 | 5 | 7.97 | 76.9 |
| comp03 | 68 | 16 | 72 | 5 | 5 | 8.17 | 78.4 |
| comp04 | 57 | 18 | 79 | 5 | 5 | 5.42 | 81.9 |
| comp05 | 139 | 9 | 54 | 6 | 6 | 21.7 | 59.6 |
| comp06 | 70 | 18 | 108 | 5 | 5 | 5.24 | 78.3 |
| comp07 | 77 | 20 | 131 | 5 | 5 | 4.48 | 80.8 |
| comp08 | 61 | 18 | 86 | 5 | 5 | 4.52 | 81.7 |
| comp09 | 75 | 18 | 76 | 5 | 5 | 6.64 | 81 |
| comp10 | 67 | 18 | 115 | 5 | 5 | 5.3 | 77.4 |
| comp11 | 13 | 5 | 30 | 9 | 5 | 13.8 | 94.2 |
| comp12 | 150 | 11 | 88 | 6 | 6 | 13.9 | 57 |
| comp13 | 66 | 19 | 82 | 5 | 5 | 5.16 | 79.6 |
| comp14 | 60 | 17 | 85 | 5 | 5 | 6.87 | 75 |
| comp15 | 68 | 16 | 72 | 5 | 5 | 8.17 | 78.4 |
| comp16 | 71 | 20 | 108 | 5 | 5 | 5.12 | 81.5 |
| comp17 | 70 | 17 | 99 | 5 | 5 | 5.49 | 79.2 |
| comp18 | 52 | 9 | 47 | 6 | 6 | 13.3 | 64.6 |
| comp19 | 66 | 16 | 74 | 5 | 5 | 7.45 | 76.4 |
| comp20 | 78 | 19 | 121 | 5 | 5 | 5.06 | 78.7 |
| comp21 | 78 | 18 | 94 | 5 | 5 | 6.09 | 82.4 |

Tabela 4.1: Tabela com informações sobre cada instância do ITC-2007

vado que separadamente eles produzem melhores soluções e de forma mais rápida. Cada um ocorre com a mesma probabilidade de 50%.

- Temperatura inicial do SA para busca local

4.4 Análise dos resultados

5 *Conclusões e trabalhos futuros*

“Nada se cria, nada se perde, tudo se transforma.”

Lavousier

Neste capítulo é apresentado as conclusões e alguns trabalhos futuros ...

5.1 Conclusões

Alguns itens interessantes para a conclusão de um projeto de graduação

Qual foi o resultado do seu trabalho? melhora na área, testes positivos ou negativos? Você acha que o mecanismo gerado produziu resultados interessantes? Quais os problemas que você encontrou na elaboração do projeto? E na implementação do protótipo? Que conclusão você tirou das ferramentas utilizadas? (heurísticas, prolog, ALE, banco de dados). Em que outras áreas você julga que este trabalho seria interessante de ser aplicado? Que tipo de continuidade você daria a este trabalho? Que tipo de conhecimento foi necessário para este projeto de graduação? Para que serviu este trabalho na sua formação?

Referências Bibliográficas

- [Achá e Nieuwenhuis 2010]ACHÁ, R. A.; NIEUWENHUIS, R. Curriculum-based course timetabling with sat and maxsat. *Annals of Operations Research*, Springer Netherlands, p. 1–21, 2010. ISSN 0254-5330. 10.1007/s10479-012-1081-x. Disponível em: <<http://dx.doi.org/10.1007/s10479-012-1081-x>>.
- [Burke et al. 2008]BURKE, E. K. et al. A branch-andcut procedure for the udine course timetabling. In: *Problem, Proceedings of the 7th PATAT Conference, 2008*. [S.l.: s.n.], 2008.
- [Ceschia, Gaspero e Schaerf 2011]CESCHIA, S.; GASPERO, L. D.; SCHAERF, A. Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem. *Computers & Operations Research*, v. 39, n. 7, p. 1615–1624, 2011. ISSN 0305-0548.
- [Chahal e Werra 1989]CHAHAL, N.; WERRA, D. D. An interactive system for constructing timetables on a pc. *European Journal of Operational Research*, Elsevier, v. 40, n. 1, p. 32–37, 1989.
- [Dinkel, Mote e Venkataramanan 1989]DINKEL, J.; MOTE, J.; VENKATARAMANAN, M. Or practice - an efficient decision support system for academic course scheduling. *Operations Research*, INFORMS, v. 37, n. 6, p. 853–864, 1989.
- [Filho e Lorena 2006]FILHO, G. R.; LORENA, L. An integer programming model for the school timetabling problem. In: CITESEER. *XIII CLAIO: Congreso Latino-Iberoamericano de Investigación Operativa*. [S.l.], 2006.
- [Goltz e Matzke 1999]GOLTZ, H.-J.; MATZKE, D. University timetabling using constraint logic programming. In: *PRACTICAL ASPECTS OF DECLARATIVE LANGUAGES*. [S.l.]: Springer-Verlag, 1999. p. 320–334.
- [Gotlieb 1962]GOTLIEB, C. C. The construction of class-teacher time-tables. In: *IFIP Congress*. [S.l.: s.n.], 1962. p. 73–77.
- [Gueret et al. 1995]GUERET, C. et al. *Building University timetables using Constraint Logic Programming*. 1995.
- [IBM 2012]IBM. *IBM ILOG CPLEX Optimization Studio*. 2012. Disponível em: <<http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/>>.
- [Insecure org 2002]Insecure org. *The Network Exploitation Tool and Security Scanner*. 2002. URL: <http://www.insecure.org/nmap>. Last Visited: 22/07/2002.
- [Junginger 1986]JUNGINGER, W. Timetabling in Germany – A survey. *Interfaces*, v. 16, n. 4, p. 66–74, 1986.

- [Lach e Lubbecke 2010]LACH, G.; LUBBECKE, M. E. Curriculum based course timetabling: new solutions to udine benchmark instances. *Annals of Operations Research*, 2010.
- [Neufeld e Tartar 1974]NEUFELD, G. A.; TARTAR, J. Graph coloring conditions for the existence of solutions to the timetable problem. *Commun. ACM*, ACM, New York, NY, USA, v. 17, n. 8, p. 450–453, ago. 1974. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/361082.361092>>.
- [Ostermann e Werra 1982]OSTERMANN, R.; WERRA, D. Some experiments with a timetabling system. *Operations-Research-Spektrum*, Springer-Verlag, v. 3, p. 199–204, 1982. ISSN 0171-6468. Disponível em: <<http://dx.doi.org/10.1007/BF01719788>>.
- [Papoulias 1980]PAPOULIAS, D. B. The assignment-to-days problem in a school time-table, a heuristic approach. *European Journal of Operational Research*, v. 4, n. 1, p. 31–41, 1980. Disponível em: <<http://EconPapers.repec.org/RePEc:eee:ejores:v:4:y:1980:i:1:p:31-41>>.
- [PATAT 2008]PATAT. *International Timetabling Competition*. 2008. URL: <http://www.cs.qub.ac.uk/itc2007>.
- [Saint Corporation 2002]Saint Corporation. *Security Administrator's Integrated Tool*. 2002. URL: <http://www.saintcorporation.com/index.html>. Last Visited: 22/07/2002.
- [Schaerf 1995]SCHAERF, A. A survey of automated timetabling. *ARTIFICIAL INTELLIGENCE REVIEW*, v. 13, p. 87–127, 1995.
- [Selim 1988]SELIM, S. M. Split vertices in vertex colouring and their application in developing a solution to the faculty timetable problem. *Comput. J.*, Oxford University Press, Oxford, UK, v. 31, n. 1, p. 76–82, fev. 1988. ISSN 0010-4620. Disponível em: <<http://dx.doi.org/10.1093/comjnl/31.1.76>>.
- [The Nessus Project 2002]The Nessus Project. *The Nessus Project*. 2002. URL: <http://www.nessus.org>. Last Visited: 22/07/2002.
- [Werra 1985]WERRA, D. de. An introduction to timetabling. *European Journal of Operational Research*, v. 19, n. 2, p. 151–162, 1985. Disponível em: <<http://EconPapers.repec.org/RePEc:eee:ejores:v:19:y:1985:i:2:p:151-162>>.

ANEXO A – Ferramentas utilizadas

Foi feita uma análise de algumas ferramentas que são muito usadas por atacantes (hackers) para a confecção de ataques. Estas ferramentas são muito úteis em vários aspectos, tais como: (1) o levantamento de informações sobre o alvo, (2) que tipo de serviços estão disponíveis no alvo, (3) quais as possíveis vulnerabilidades do alvo, entre outras informações. As ferramentas analisadas foram o *nmap* [Insecure org 2002], o *nessus* [The Nessus Project 2002], o *saint* [Saint Corporation 2002], além de alguns comandos de sistemas operacionais (UNIX-Like e Windows-Like) usados para rede, tais como o *ping*, *nslookup* e *whois*.