

# Plano de Trabalho de Conclusão de Curso

## Proposta de um algoritmo híbrido para solução do Problema de Escalonamento de Tripulação (CSP)

Renan Samuel da Silva – `renan.samuel.da.silva@gmail.com`  
Omir Alves – `omir.alves@udesc.br` (*orientador*)

Turma 2016/2 – Joinville/SC

02 de Julho de 2016

### Resumo

Grande parte dos problemas encontrados na prática são difíceis de serem resolvidos e possuem um grande número de variáveis (na ordem dos milhares ou milhões), fazendo com que seja impossível resolver ótimamente em uma escala de tempo plausível. Sendo assim, torna-se necessário o emprego de algoritmos cada vez mais complexos e que unem soluções exatas e heurísticas. O objetivo deste trabalho é pesquisar, identificar e especificar um método para resolver o problema de agendamento de tripulação (*crew scheduling*).

**Palavras-chave:** *Programação linear inteira, otimização combinatoria, cobertura de conjunto, agendamento de tripulação, Geração de Colunas*

## 1 Introdução e Justificativa

Cada vez mais as empresas buscam otimizar suas atividades, de modo a reduzir o custo e maximizar o lucro final. As empresas do segmento de transporte urbano de ônibus deparam-se com o desafio que precisam ser abordados a fim de viabilizar o negócio, dentre as quais pode-se citar: quais veículos da frota devem ser renovados; quando realizar manutenção preventiva na frota; quais horários do serviço de ônibus devem ser disponibilizados; identificar as rotas do serviço de transporte; como alocar os funcionários; como tratar imprevistos temporais e climáticos, dentre outras. Estes questionamentos podem ser resolvidos e otimizados através da utilização de recursos de pesquisa operacional.

Especificamente o problema de alocação de funcionários ou tripulação, em inglês *crew scheduling problem* (CSP), consiste em escolher quais funcionários devem realizar quais funções durante um dado período de tempo [Bergh et al. 2013]. O processo de escolha está sujeito à diversas restrições. Existem restrições devido a regras definidas pela empresa, leis trabalhistas e sindicais, e até mesmo preferências pessoais dos motoristas. Dado o conjunto de restrições, deve-se encontrar as designações que reduzem ao máximo o custo da operação. O CSP assume que já tenha sido determinado as rotas onde a tripulação irá trabalhar, assim como quais veículos serão utilizados, quais os horários de partida e chegada e os pontos de troca.

A definição de muitos termos relacionados à problemática do CSP faz-se necessária e são apresentadas a seguir: uma **viagem** consiste no deslocamento entre dois pontos pré-determinados com horários de partida e chega já determinados. Uma **jornada** consiste na sequência de viagens realizadas por uma dada tripulação durante o seu turno de trabalho. Conhecendo todas as possíveis jornadas válidas sobre o conjunto de regras pré-estabelecidos, a solução do CSP consiste em escolher quais jornadas cobrem todas as viagens que devem ser realizadas [Bergh et al. 2013]. Ou seja, dado o conjunto de jornadas válidas, escolhe-se um subconjunto que cubra todas as viagens pelo menos uma vez, minimizando assim o custo final.

O problema do CSP é referenciado na literatura como sendo o problema de **cobertura de conjuntos**, ou *set covering problem* (SCP) [Nemhauser e Wolsey 1988]. Onde dado um conjunto de linhas a serem cobertas, e um conjunto de colunas com custos associados que cobrem as linhas, escolhe-se o subconjunto de colunas que cubra todas as linhas minimizando o custo final. Como o SCP define que cada linha deve ser coberta pelo menos uma vez, isso implica que se o CSP for modelado como uma instância de SCP, pode ocorrer de mais de um tripulante ser designado para a mesma viagem. Isso na prática consiste de um motorista ir de carona até o início de outra viagem, por exemplo.

Outra forma de modelar-se o CSP consiste em determinar que exatamente uma tripulação deve ser associado a uma viagem. Esta restrição transforma o problema de SCP em um problema de **particionamento de conjuntos**, *set partitioning problem* (SPP) [Garfinkel e Nemhauser 1969]. A principal diferença entre o SCP e o SPP consiste a impossibilidade de ter uma mesma linha coberta por mais de uma coluna. No contexto do CSP, isto implica em não ter uma tarefa sendo coberta por mais de uma jornada. Devido a restrição de não sobreposição, um CSP modelado como um SPP pode ser significativamente mais complexo de resolver, porém, mais interessante do ponto de vista prático, já que diminui a ociosidade dos tripulantes. Sabe-se que tanto o SPP quanto o SCP são problemas NP-Hard [Karp 1972]. Note que toda solução do SPP é uma solução para o SCP, porém o recíproco não é verdadeiro.

Um dos primeiros algoritmos propostos para o SCP consiste em uma heurística gulosa proposta por [Chvatal 1979]. A cada passo o algoritmo escolhe a coluna que cobre o maior número possível de linhas. O algoritmo é na prática rápido, porém tende a não gerar soluções tão boas quanto outros algoritmos modernos. [Balas e Ho 1980] propôs um algoritmo baseado em *branch and bound* e que utiliza heurísticas duais. Este algoritmo (com o poder computacional disponível na data da publicação do artigo) foi capaz de resolver instâncias de dimensões até  $200 \times 2000$ . [Beasley 1987] melhorou o algoritmo proposto por Balas e Ho, utilizando relaxações lagrangeanas e remoção de linhas e colunas. Este algoritmo, segundo o autor, chegou a marca de problemas de dimensões  $400 \times 4000$ . [Fisher e Kedia 1990] utilizou *branch and bound* com diversas heurísticas duais para assim encontrar o limite superior de otimalidade do problema. Trabalhos como [Freitas 2011] e [Santos 2008] utilizam soluções híbridas para a solução do problema. [Ceria, Nobili e Sassano 1997], [Caprara, Toth e Fischetti 2000], [Ernst et al. 2004] e [Bergh et al. 2013] apresentam um estudo mais detalhado sobre os algoritmos utilizados para resolver o SCP, SPP e CSP.

Conhecendo-se todas as possíveis jornadas de uma instância de CSP, pode-se modela-lo utilizando programação linear inteira (PLI). A modelagem utilizando-se o SPP e o SCP é apresentada em (1) e (4), respectivamente. O conjunto de todas as jornadas possíveis está codificado na matrix  $A$ . O vetor  $J$  corresponde a todas as jornadas, e o vetor  $T$  a todas as tarefas. A variável  $a_{tj}$  é 1 se a tarefa  $t$  é coberta pela jornada  $j$  e 0 caso contrário. A restrição (2) garante que cada tarefa  $t \in T$  é coberta exatamente uma única vez pelas jornadas selecionadas.

A restrição (5) funciona de modo análogo, porém, a restrição deixa de ser uma igualdade e passa a ser uma desigualdade, fazendo com que cada tarefa seja coberta pelo menos uma vez. O vetor  $X$  determina que jornadas serem utilizadas. Se  $x_j = 1$ , então a  $j$ -ésima é utilizada, caso contrário  $x_j = 0$ . As restrições (3) e (6) garantem que a variável de decisão  $x$  possuirá um valor válido.

Dado o problema em questão, o tamanho da matriz é tipicamente grande, e para casos práticos é muitas vezes inviável de tratá-la. O número de jornadas possíveis cresce exponencialmente, de modo que a enumeração de todas as jornadas possíveis não é viável. [Vance 1993] reportou que para uma instância de CSP com 253 tarefas, mais de 5 milhões de jornadas são possíveis. De todas as jornadas possíveis, apenas algumas são de fato utilizadas na solução final, as demais não são interessantes. Portanto, é interessante que apenas jornadas que podem vir a ser úteis sejam úteis de fato. A programação linear inteira dispõe-se um recurso capaz de sequencialmente reduzir o número de colunas a serem utilizados: **geração de colunas** [Desrosiers e Lübbecke 2005].

$$\min \sum_{j \in J} c_j x_j \quad (1)$$

$$\sum_{j \in J} a_{tj} x_j = 1, \forall t \in T \quad (2)$$

$$x_j \in \{0, 1\}, \forall j \in J \quad (3)$$

$$\min \sum_{j \in J} c_j x_j \quad (4)$$

$$\sum_{j \in J} a_{tj} x_j \geq 1, \forall t \in T \quad (5)$$

$$x_j \in \{0, 1\}, \forall j \in J \quad (6)$$

O método de geração de colunas consiste em um processo onde o problema com um grande número de variáveis é decomposto em dois problemas. O problema mestre, que é igual ao original, porém, com um conjunto reduzido de variáveis. E o sub-problema, que é utilizado para identificar quais variáveis são necessárias para obter-se a solução ótima do problema original. O processo de solução com geração de colunas consiste em criar-se um problema mestre, otimizar a sua relaxação linear, e utilizar as variáveis duais do problema mestre juntamente com o sub-problema para indentificar se o problema mestre necessita da geração de novas colunas ou se ele corresponde ao

ótimo do problema original.

Para a geração de colunas, dado o conjunto  $J$  de todas as jornadas (que na prática está codificado na matriz  $A$ ), utiliza-se um conjunto reduzido  $\tilde{J} \subseteq J$ . O problema mestre é dado por (7), (8) e (9). Após a solução da relaxação linear do problema mestre, utiliza-se as variáveis duais em um subproblema para determinar se alguma jornada pode melhorar o resultado atual. Este processo é denominado de *pricing*. No caso do CSP, pode-se apresentar o conjunto de tarefas como vértices em um grafo e liga-los com arestas direcionadas para indicar a ordem que as tarefas podem ser executadas. Toda jornada é então um caminho neste grafo. A tabela 1 contém uma instância de CSP e a figura ?? apresenta o grafo associado ao problema.

$$\min \sum_{j \in \tilde{J}} c_j x_j \quad (7)$$

$$\sum_{j \in \tilde{J}} a_{tj} x_j = 1, \forall t \in T \quad (8)$$

$$x_j \in \{0, 1\}, \forall j \in \tilde{J} \quad (9)$$

tarefa	Início	Fim
tarefa 1	1	10
tarefa 2	1	4
tarefa 3	11	14
tarefa 4	6	10
tarefa 5	11	15

Tabela 1: Instância do CSP

Alem do método de geração de colunas, outros métodos também são utilizados na literatura, tais como: algoritmos genéticos [Martins e Silva 2015], *multi-start randomized heuristic* [Armas et al. 2016], *particle swarm optimization* [Limlawan, Kasemsontitum e Jeenanunta 2014], *simulated annealing* [Hanafi e Kozan 2014] e *ant colony optimization* [Deng e Lin 2011].

Na literatura pesquisada até o momento, identificou-se que vários autores a fim de resolver o CSP de forma eficiente (que apresentam um melhor desempenho computacional), utilizaram o método de geração de colunas ou um conjunto de outros métodos, podendo ser heurísticos ou híbridos.

Porém, dada a pesquisa que pode-se realizar, não foi possível concluir que identificou-se um método (híbrido ou heurístico) que apresenta-se o melhor desempenho computacional dentre todos os métodos que foram propostos. Especialmente considerando que o CSP apresenta diversas modelagens que variam entre si, de modo que o bom desempenho para um conjunto de modelos não transfere-se para outro.

Dado o exposto acima, este trabalho tem o objetivo de especificar e propor um método para a solução do CSP que, a princípio seja capaz de prover soluções de modo exato e seja competitivo, e preferencialmente mais eficiente, com outros algoritmos pesquisados na literatura.

## 2 Objetivos

Objetivo Geral: Este trabalho tem como objetivo analisar os métodos disponíveis na literatura para solução do CSP com a finalidade de propor um novo algoritmo ou melhorar um já existente.

Objetivos Específicos:

- Pesquisar e identificar os principais algoritmos presentes na literatura que resolvam o CSP.
- Coletar instâncias de benchmark utilizadas na literatura.
- Selecionar alguns dos algoritmos pesquisados.
- Especificar o método proposto.
- Realizar testes para avaliação do método proposto.

## 3 Metodologia

Inicialmente estudar-se-á em detalhe o problema proposto e será realizado uma revisão bibliográfica do assunto, incluindo as diversas modelagens para o problema. Em seguida será considerado algoritmos que foram utilizados na literatura para resolver o problema e será analisado suas características funcionais e desempenho, com o objetivo de elencar 3 algoritmos para estudo. Após esta etapa será procurado problemas práticos reais de PPT com o objetivo de realizar um estudo da eficiência dos modelos estudados. Por fim será realizado uma comparação dos resultados obtidos para averiguar a relevância dos algoritmos estudados e suas implementações.

## 4 Cronograma proposto

Etapas:

1. Revisão bibliográfica e análise da problemática.
2. Escolha dos métodos de modelagem.
3. Estudo de algoritmos exatos aplicáveis para a solução da problemática.
4. Estudo de algoritmos heurísticos aplicáveis para a solução da problemática.
5. Eleição de algoritmos para estudo.
6. Desenvolvimento de protótipos para testes.
7. Experimentação com os algoritmos escolhidos.
8. Coleta e análise dos experimentos.
9. Escrita do trabalho de conclusão de curso.

Etapas	2012												2013											
	J	F	M	A	M	J	J	A	S	O	N	D	J	F	M	A	M	J	J	A	S	O	N	D
1																								
2																								
3																								
4																								
5																								
6																								
7																								
8																								
9																								
...																								

## 5 Forma de Acompanhamento/Orientação

O acompanhamento será realizado principalmente através de reuniões semanais. O controle das tarefas a fazer serem feitas baseadas em uma ata gerada a cada reunião.

## Referências

- [Armas et al. 2016]ARMAS, J. de et al. A multi-start randomized heuristic for real-life crew rostering problems in airlines with work-balancing goals. *Annals of Operations Research*, Springer, p. 1–24, 2016.
- [Balas e Ho 1980]BALAS, E.; HO, A. Set covering algorithms using cutting planes, heuristics, and subgradient optimization: a computational study. In: *Combinatorial Optimization*. [S.l.]: Springer, 1980. p. 37–60.
- [Beasley 1987]BEASLEY, J. E. An algorithm for set covering problem. *European Journal of Operational Research*, Elsevier, v. 31, n. 1, p. 85–93, 1987.
- [Bergh et al. 2013]BERGH, J. V. den et al. Personnel scheduling: A literature review. *European Journal of Operational Research*, v. 226, n. 3, p. 367 – 385, 2013. ISSN 0377-2217.
- [Bergh et al. 2013]BERGH, J. Van den et al. Personnel scheduling: A literature review. *European Journal of Operational Research*, Elsevier, v. 226, n. 3, p. 367–385, 2013.
- [Caprara, Toth e Fischetti 2000]CAPRARA, A.; TOTH, P.; FISCHETTI, M. Algorithms for the set covering problem. *Annals of Operations Research*, Springer, v. 98, n. 1-4, p. 353–371, 2000.
- [Ceria, Nobili e Sassano 1997]CERIA, S.; NOBILI, P.; SASSANO, A. Set covering problem. *Annotated Bibliographies in Combinatorial Optimization*, Springer, p. 415–428, 1997.
- [Chvatal 1979]CHVATAL, V. A greedy heuristic for the set-covering problem. *Mathematics of operations research*, INFORMS, v. 4, n. 3, p. 233–235, 1979.
- [Deng e Lin 2011]DENG, G.-F.; LIN, W.-T. Ant colony optimization-based algorithm for airline crew scheduling problem. *Expert Systems with Applications*, Elsevier, v. 38, n. 5, p. 5787–5793, 2011.

- [Desrosiers e Lübbecke 2005] DESROSIERS, J.; LÜBBECKE, M. E. A primer in column generation. In: *Column generation*. [S.l.]: Springer, 2005. p. 1–32.
- [Ernst et al. 2004] ERNST, A. T. et al. Staff scheduling and rostering: A review of applications, methods and models. *European journal of operational research*, Elsevier, v. 153, n. 1, p. 3–27, 2004.
- [Fisher e Kedia 1990] FISHER, M. L.; KEDIA, P. Optimal solution of set covering/partitioning problems using dual heuristics. *Management science*, INFORMS, v. 36, n. 6, p. 674–688, 1990.
- [Freitas 2011] FREITAS, A. de. *UM ALGORITMO HÍBRIDO PARA A SOLUÇÃO DE PROBLEMAS DE ESCALA DE MOTORISTAS*. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, 2011.
- [Garfinkel e Nemhauser 1969] GARFINKEL, R. S.; NEMHAUSER, G. L. The set-partitioning problem: set covering with equality constraints. *Operations Research*, INFORMS, v. 17, n. 5, p. 848–856, 1969.
- [Hanafi e Kozan 2014] HANAFI, R.; KOZAN, E. A hybrid constructive heuristic and simulated annealing for railway crew scheduling. *Computers & Industrial Engineering*, Elsevier, v. 70, p. 11–19, 2014.
- [Karp 1972] KARP, R. M. Reducibility among combinatorial problems. Springer, p. 85–103, 1972.
- [Limlawan, Kasemsontitum e Jeenanunta 2014] LIMLAWAN, V.; KASEMSONTITUM, B.; JEENANUNTA, C. A hybrid particle swarm optimization and an improved heuristic algorithm for an airline crew rostering problem. *SCIENCEASIA*, THAILANDS NATL SCIENCE & TECHNOLOGY DEVELOPMENT AGENCY PUBLIC INFORMATION DEPT, 73/1 RAMA VI RD, RAJDHEVEE, BANGKOK, 00000, THAILAND, v. 40, n. 6, p. 456–462, 2014.
- [Martins e Silva 2015] MARTINS, L. do C.; SILVA, G. P. Algoritmos genéticos aplicados ao problema de rodízio de tripulações. *Simpósio Brasileiro de Pesquisa Operacional*, 2015.
- [Nemhauser e Wolsey 1988] NEMHAUSER, G. L.; WOLSEY, L. A. Integer programming and combinatorial optimization. Wiley, Chichester. *GL Nemhauser, MWP Savelsbergh, GS Sigismondi (1992). Constraint Classification for Mixed Integer Programming Formulations. COAL Bulletin*, Springer, v. 20, p. 8–12, 1988.
- [Santos 2008] SANTOS, A. G. dos. *Método de Geração de Colunas e Meta-heurísticas para Alocação de Tripulação*. Tese (Doutorado) — Universidade Federal de Minas Gerais, 2008.
- [Vance 1993] VANCE, P. H. Crew scheduling, cutting stock, and column generation: Solving huge integer programs. Georgia Institute of Technology, 1993.