

# Estratégias de modelagem para o Problema do Caixeiro Viajante utilizando o software CPLEX.

Maria Eduarda B. Lacerda Gomes<sup>1</sup>, Ohana Paula Lopes<sup>2</sup>,  
Fermín A. Tang Montane<sup>2</sup>

<sup>1</sup>Laboratório de Engenharia de Produção (LEPROD) – Universidade Estadual do Norte Fluminense (UENF) – Campos dos Goytacazes - RJ

<sup>2</sup>Laboratório de Ciências Matemáticas (LCMAT) - Universidade Estadual do Norte Fluminense (UENF) – Campos dos Goytacazes - RJ

lacerda.me@gmail.com, ohanap.lobes@gmail.com, tang@uenf.br

**Abstract:** *In this work two mathematical models for the Traveling Salesman Problem (TSP) were implemented using the modeling language of CPLEX software, with the aim to compare computational performance of both models. Computational tests were conducted to compare two different methodologies for the modeling of combinatorial optimization problems. On one side, one model is generated in a static way, with all its restrictions, on the other side, the other model is generated in a dynamic way, by the addition of restrictions as needed. Tests showed that the second approach offer significant computational gains.*

## 1. Introdução

O Problema do Caixeiro Viajante (PCV) - *Traveling Salesman Problem (TSP)* - é um problema clássico de programação inteira. Nesse problema, são dadas  $n$  cidades e as distâncias entre elas. O objetivo do problema é encontrar uma sequência de visita às cidades de maneira que o caixeiro visite todas elas, uma única vez, percorrendo a menor distância possível. Considera-se o caso simétrico, em que a distância de ida de uma cidade a outra é igual à distância de volta. Sabe-se que problemas de programação inteira podem ser modelados de formas diferentes, sendo que o modelo influencia na eficiência do método de solução usado para resolvê-lo. No presente trabalho, comparam-se dois modelos para resolver o problema PCV. Os modelos foram implementados na linguagem OPL do software CPLEX, e resolvidos usando o método *branch and cut* disponível nesse software.

## 2.- Metodologia

O trabalho consistiu em avaliar dois modelos para o problema PCV. A formulação proposta por Miller *et. al.* (1960) e a formulação de eliminação de *subtours* proposta por Dantzig *et. al.* (1954) [Goldberg e Luna 2005]. O primeiro modelo foi implementado na linguagem OPL do software CPLEX [IBM Corp 2009], enquanto o segundo faz parte da biblioteca disponível nesse software. Observa-se que as restrições de eliminação de *subtours* crescem exponencialmente com o tamanho do problema. O objetivo do teste foi avaliar o esforço computacional ao resolver o problema PCV usando duas estratégias de solução diferentes decorrentes dos modelos mencionados. No primeiro caso, a estratégia tradicional define um único problema que é resolvido pelo método *branch and cut*. Tal método explora o espaço de soluções, fixando o valor

de algumas variáveis, testando combinações de maneira eficiente e reduzindo o espaço de busca, descartando as regiões que contêm soluções sub-ótimas. A estratégia no segundo modelo, define inicialmente um problema menor, com um subconjunto pequeno de restrições, que é resolvido e estendido gradativamente pela adição dinâmica de restrições violadas, resolvendo-se assim, vários subproblemas. Considera-se inicialmente um problema relaxado, compreendendo apenas um subconjunto do total de restrições. Dessa forma, o espaço de busca é consideravelmente menor ao do problema original, sendo que o problema relaxado pode ser resolvido mais rapidamente pelo método *branch and cut*. Após a resolução do problema relaxado, identifica-se se existem restrições violadas. Em caso afirmativo, um novo problema relaxado é gerado e resolvido, incorporando-se tais restrições. O procedimento é repetido até que nenhuma nova restrição seja violada, o que garante a obtenção da solução ótima.

### 3.- Resultados

Ambos os modelos foram testados utilizando instâncias teste com: 6, 10, 17 e 50 cidades. Os resultados dos testes se encontram na Tabela 1. Observa-se que para um problema com  $n$  cidades, o primeiro modelo requer  $n^2$  variáveis, enquanto o segundo modelo utiliza  $(n^2 - n) / 2$  variáveis. Por outro lado, enquanto o primeiro modelo resolve apenas um problema, o segundo modelo resolve vários subproblemas. Com relação ao desempenho computacional, observa-se que o segundo modelo teve, em geral, melhor desempenho. Nos testes realizados, observou-se uma diferença significativa na instância teste com 50 cidades.

**Tabela 1: Testes realizados no software CPLEX para o problema PCV.**

Total de Cidades (n)	Formulação de Miller et. al. (1960)			Formulação de Dantzig et. al. (1954)		
	Número de Variáveis	Número de Subprocessos	Tempo (segs.)	Número de Variáveis	Número de Subprocessos	Tempo (segs.)
6	36	1	0,20	15	1	0,09
10	100	1	0,18	45	1	0,09
17	289	1	1,62	136	25	2,38
50	2500	1	18,5	1225	8	1,48

### 4. Conclusão

Foram comparados dois modelos de programação matemática para resolver o problema do caixeiro viajante. Observou-se que, enquanto que o primeiro modelo gera todas as restrições uma única vez, gerando um único problema, o segundo modelo gera restrições de maneira dinâmica criando vários subproblemas, pequenos se comparados ao problema original. A convergência do primeiro modelo tende a ser lenta enquanto que o segundo, apesar de gastar mais tempo na geração de subproblemas, mostrou ser mais eficiente. Testes adicionais serão realizados em problemas de maior porte e se estudarão aplicações desta estratégia a outros problemas de otimização combinatória.

### 5. Referências

- Goldberg, M.C. e Luna, H.P.L. (2005), Otimização combinatória e programação linear, Editora Elsevier, 2<sup>a</sup> Edição.
- IBM Corp. (2009), IBM ILOG CPLEX v12.1 User's Manual for CPLEX, International Business Machine Corporation.