

**Universidade Federal de Ouro Preto - UFOP**  
**Instituto de Ciências Exatas e Biológicas - ICEB**  
**Departamento de Computação - DECOM**

## **Um Algoritmo Heurístico Aplicado a Produção em Sistemas de Manufatura Flexíveis**

**Bolsista:** Túlio Neme de Azevedo

**Orientador:** Prof. Dr. Marco Antonio Moreira de Carvalho – DECOM/UFOP

**Nota:** Relatório Semestral referente ao período de 01/03/2016 a 31/07/2016, apresentado à Universidade Federal de Ouro Preto, como parte das exigências do programa PIVIC.

**Local:** Ouro Preto - Minas Gerais - Brasil

**Data:** 25 de agosto de 2016

# **Um Algoritmo Heurístico Aplicado a Produção em Sistemas de Manufatura Flexíveis**

Problema de minimização de troca de ferramentas é um problema combinatório encontrado nas linhas de produção, que destina um conjunto tarefas a uma única máquina flexível, que possui uma capacidade ferramentas fixa. Cada tarefa requer um conjunto de ferramentas específicas para o seu devido processamento. Durante o processamento das tarefas, trocas de ferramentas são necessárias, respeitando a capacidade máxima da máquina, para processar todas as tarefas. Assim, é necessário sequenciar as tarefas de forma a minimizar o número de trocas de ferramentas durante o processamento de todas as tarefas alocadas para a máquina. Foi realizado uma revisão bibliográfica sobre o tema para servir de base para a implementação de um algoritmo heurístico para resolução do problema. Testes e experimentos computacionais serão realizados sobre instâncias presentes na literatura.

Assinatura do orientador(a): \_\_\_\_\_  
Marco Antonio Moreira de Carvalho

Assinatura do bolsista: \_\_\_\_\_  
Túlio Neme de Azevedo

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contextualização . . . . .	1
1.2	Descrição do Problema . . . . .	2
<b>2</b>	<b>Objetivos</b>	<b>5</b>
<b>3</b>	<b>Fundamentos Teóricos e Revisão</b>	<b>6</b>
<b>4</b>	<b>Materiais e Métodos</b>	<b>9</b>
4.1	Modelagem do Problema . . . . .	9
4.2	O Resolvedor Concorde . . . . .	11
4.3	O Problema de Minimização de Trocas de Ferramentas e o Problema do Caixa- xeiro Viajante . . . . .	11
<b>5</b>	<b>Conclusão</b>	<b>14</b>

# Capítulo 1

## Introdução

Neste capítulo é apresentado uma introdução ao tema abordado neste trabalho, bem como uma descrição formal do Problema de Minimização de Trocas de Ferramentas, com exemplos.

### 1.1 Contextualização

Com a evolução frequente da tecnologia, as indústrias que utilizam máquinas flexíveis em suas linhas de produção recorrem a métodos computacionais para resolverem problemas operacionais. Por sua vez, esses métodos, com suas respectivas especificidades, podem contribuir significativamente para otimização da eficiência das linhas de produção.

Uma *máquina flexível* tem como característica proporcionar dinamismo à linha de produção ao fabricar diversos tipos de produtos não relacionados entre si. Este tipo de máquina comporta um número fixo de *ferramentas* instaladas, por exemplo: chaves de fenda, alicates e brocas entre outras. Diferentes conjunto de ferramentas produzem diferentes tipos de produtos. Cada produto exige que um determinado conjunto de ferramentas esteja instalado no momento de sua produção.

A capacidade da máquina de produção é suficiente para suportar o conjunto de ferramentas necessárias para a fabricação de cada produto isoladamente, porém não comporta todas as ferramentas necessárias para fabricar todos os produtos simultaneamente. Desta maneira cada troca exige que a máquina seja desligada, interrompendo a linha de produção, ocasionando prejuízo, aumento de custo de produção, lentidão, etc.

Ao iniciar a fabricação dos produtos, trocas de ferramentas serão necessárias, para que a capacidade não seja ultrapassada e o produto subsequente seja processado. Assim, ao trocar as ferramentas da máquina, a linha de produção é interrompida inteiramente, gerando perda de tempo. Um dos problemas que está presente nas indústrias que empregam máquina flexíveis, é o *Problema de Minimização de Troca de Ferramentas* (MTSP), um problema de otimização combinatória. Este problema consiste em determinar a sequência de fabricação dos produtos a cada estágio nas linhas de produção. A fabricação de um produto é considerada uma *tarefa* a ser realizada por uma máquina flexível. O objetivo do MTSP é otimizar a linha de produção pela minimização das trocas de ferramentas necessárias. Desta forma, a produção é interrompida minimamente, tornando-a mais eficiente.

O MTSP se divide em duas partes: o problema de carregamento de ferramentas e o problema de sequenciamento de tarefas, como definido por Tang and Denardo (1988). O problema de carregamento de ferramentas, dada uma sequência fixa de tarefas, consiste em determinar o menor número de trocas de ferramentas para processar o conjunto. O método *Keep Tool Needed Soonest* (KTNS) (Tang and Denardo, 1988) soluciona este problema em tempo deter-

minístico polinomial. Quando troca de ferramentas são necessárias, a política KTNS garante que as ferramentas que serão necessárias mais brevemente pelas tarefas ainda não realizadas sejam mantidas na máquina. O problema de sequenciamento de tarefas é  $\mathcal{NP}$ -Difícil (Crama et al., 1994), significando que não se conhece algoritmo eficiente para sua solução. Sendo assim se faz necessário resolver o problema de sequenciamento das tarefas.

O MTSP se apresenta de diferentes formas na literatura. Este trabalho aborda o caso geral, no qual as ferramentas possuem tamanho e custo uniformes, ou seja, não importa qual a posição as ferramentas são alocadas na máquina flexível e o tempo de instalação é o mesmo para todas ferramentas.

## 1.2 Descrição do Problema

Formalmente, o MTSP é definido da seguinte forma: dados uma máquina flexível com capacidade de comportar até  $C$  ferramentas, um conjunto de tarefas  $T = [1, \dots, n]$ , um conjunto de ferramentas  $F = [1, \dots, m]$ , o subconjunto de ferramentas  $F_i$  ( $F_i \in F$ ) necessárias para realizar a tarefa  $i$  ( $i \in T$ ), é necessário determinar uma permutação  $\phi$  dos elementos de  $T$  tal que o número de trocas de ferramentas, obtido pela aplicação do algoritmo KTNS, seja minimizado.

Uma instância do MTSP apresenta informações sobre as ferramentas necessárias para execução de cada uma das tarefas. A Tabela 1.1 apresenta os dados de um cenário relacionado ao MTSP em que tem-se  $n = 5$ ,  $m = 5$ ,  $C = 3$ .

Tarefas	Ferramentas
1	2, 3, 5
2	1, 3
3	1, 4, 5
4	1, 2
5	2, 3, 4

Tabela 1.1: Exemplo de instância MTSP.

Na referida Tabela 1.1, a primeira coluna representa as tarefas a serem processadas pela máquina flexível, enumeradas de 1 a 5. A segunda coluna representa as ferramentas necessárias para que a tarefa referida seja processada. Computacionalmente, esta instância pode ser modelada por uma matriz binária  $Q$ , conforme apresentado pela Tabela 1.2. As linhas representam as ferramentas e as colunas representam as tarefas, ambas numeradas de 1 a 5. Os elementos  $q_{ij}$  da matriz  $Q$  são preenchidos da seguinte maneira:  $q_{ij} = 1$  caso a ferramenta  $i$  ( $i \in F$ ) for necessária para processar a tarefa  $j$  ( $j \in T$ ). Caso contrário,  $q_{ij} = 0$ .

Ferramentas \ Tarefas	1	2	3	4	5
1	0	1	1	1	0
2	1	0	0	1	1
3	1	1	0	0	1
4	0	0	1	0	1
5	1	0	1	0	0

Tabela 1.2: Matriz  $Q$ .

Modelado o problema, uma solução é dada por um sequenciamento  $\phi$  das colunas de  $Q$ . Por exemplo, considere  $\phi = [3, 4, 2, 1, 5]$ , representada pela Tabela 1.3. Esta solução resulta em 7 troca de ferramentas, explicitadas a seguir:

Tarefas	Ferramentas carregadas na máquina
3	1, 4, 5
4	1, 2, 5
2	1, 2, 3
1	2, 3, 5
5	2, 3, 4

Tabela 1.3: Solução MTSP.

1. Três trocas para carregar as ferramentas iniciais na máquina (ferramentas 1,4,5);
2. Uma troca entre a tarefa 3 e 4 (ferramenta 4 por 2);
3. Uma troca entre a tarefa 4 e 2 (ferramenta 5 por 3);
4. Uma troca entre a tarefa 2 e 1 (ferramenta 1 por 5); e
5. Uma troca entre a tarefa 1 e 5 (ferramenta 5 por 4).

Considerando a representação por matrizes binárias, a solução  $\phi$  induz uma matriz permutação  $R^\phi$ , vide Tabela 1.4, cujas colunas são as colunas de  $Q$  na ordem estabelecida por  $\phi$  e uma coluna adicional 0 com todos os elementos nulos. Na referida Tabela 1.4, as ferramentas inseridas a cada instante na máquina estão sublinhadas e as ferramentas que permanecem na máquina, mesmo sem serem utilizadas, estão indicadas em negrito. Por exemplo, na troca da tarefa 3 para a tarefa 4, a ferramenta 2 foi inserida na máquina, portanto sublinhada, e a ferramenta 5 foi mantida na máquina mesmo sem ser utilizada na tarefa 4, portanto em negrito.

Ferramentas \ $\phi$	0	3	4	2	1	5
1	0	<u>1</u>	1	1	0	0
2	0	0	<u>1</u>	<b>1</b>	1	1
3	0	0	0	<u>1</u>	1	1
4	0	<u>1</u>	0	0	0	<u>1</u>
5	0	<u>1</u>	<b>1</b>	0	<u>1</u>	0

Tabela 1.4: Matriz  $R^\phi$ .

O valor dos elementos  $r^\phi$  são definidos de acordo com a Equação 1.2.

$$r_{ij}^\phi = \begin{cases} 1, & \text{se a ferramenta } i \in F \text{ estiver na máquina durante o processamento da tarefa } j \in T \\ 0, & \text{caso contrário.} \end{cases}$$

Uma solução representada pela matriz  $R^\phi$  é avaliada de acordo com a Equação 1.1, proposta por Crama et al. (1994). Esta função calcula, dada uma sequência de tarefas, o número de inversões de 0 para 1, que representam a inserção de ferramentas na máquina, por exemplo, vide os dados contidos na Tabela 1.4, cujo número de trocas é 7.

$$Z_{MTSP}^\phi(R) = \sum_{j \in T} \sum_{i \in F} r_{ij}^\phi (1 - r_{i,j-1}^\phi) \quad (1.1)$$

O objetivo do MTSP é determinar a permutação  $\phi \in \Phi$  das colunas da matriz  $Q$  que resulte no menor número de trocas de ferramentas, em que  $\Phi$  é o conjunto de todas as permutações possíveis. A função objetivo correspondente é exibida na Equação 1.2.

$$\min_{\phi \in \Phi} Z_{MTSP}^{\phi}(Q) \quad (1.2)$$

Conforme mencionado anteriormente, o Problema de Minimização de Trocas de Ferramentas pertence à classe  $\mathcal{NP}$ -Difícil para os casos em que  $C \geq 2$  (Crama et al., 1994).

# Capítulo 2

## Objetivos

O trabalho de pesquisa abordado neste projeto se propõe a estudar o Problema de Minimização de Trocas de Ferramentas em sistemas de manufatura flexíveis e a desenvolver abordagens heurísticas consistentes, robustas e que constituam uma contribuição para o estado da arte relacionado ao problema tratado e correlatos.

Objetivos específicos:

1. Realizar pesquisa para geração de embasamento teórico e revisão bibliográfica sobre o tema tratado;
2. Propor um método heurístico que contemple apropriadamente as especificidades do problema tratado;
3. Avaliar o método implementado considerando dados reais e também com problemas teste publicamente disponíveis, realizando uma análise crítica considerando outros métodos da literatura;
4. Buscar a aplicação prática dos métodos desenvolvidos em contextos reais, a fim de que também seja constituído um avanço para as indústrias nacionais.



## Capítulo 3

# Fundamentos Teóricos e Revisão

Os conceitos básicos deste trabalho foram fundamentados através dos autores com os respectivos trabalhos referentes ao MTSP mencionados a seguir. A revisão da literatura é apresentada em ordem cronológica.

Tang and Denardo (1988) foram os primeiros a relataram na literatura uma abordagem ao MTSP, utilizando modelagem por grafos. Porém, a principal contribuição deste trabalho foi a política ótima *Keep Tool Needed Soonest* (KTNS) para troca de ferramentas. Dada uma sequência fixa de tarefas a serem realizadas, esta política define o menor número de troca de ferramentas correspondente. Se trocas forem necessárias, a política garante que as ferramentas que serão necessárias mais brevemente pelas tarefas ainda não realizadas sejam mantidas na máquina flexível.

No mesmo ano, Bard (1988) formulou o MTSP como um problema de programação inteira não linear e o resolveu com uma heurística baseada na relaxação dual do modelo. O conjunto de instâncias utilizados nos experimentos não teve sua origem relatada. Todavia, os resultados demonstraram a necessidade de um tempo de execução relativamente pequeno para determinar boas soluções locais.

Crama et al. (1994) relacionaram o MTSP com o Problema do Caixeiro Viajante (PCV), o que inovou a modelagem do MTSP. Neste trabalho foram aplicadas heurísticas específicas para o PCV na resolução do MTSP. Também foi definida a complexidade computacional do problema:  $\mathcal{NP}$ -Difícil para  $C \geq 2$ . Posteriormente, nesta mesma modelagem, Hertz et al. (1998) embutiram a política KTNS em heurísticas existentes para solução do PCV, duas heurísticas de Gendreau et al. (1992), *GENI*, e *GENIUS*, e uma de Golden and Stewart (1985), *Farthest Insertion*. A heurística *GENI* consiste em um método construtivo, ou seja, gera soluções gradativamente, baseando-se em métodos de inserção. A heurística *GENIUS* utiliza o mesmo princípio de *GENI*, porém, com uma fase de pós-otimização, denominadas *US*. O método *Farthest Insertion* tem como característica inserir na solução o elemento cuja distância seja a maior em relação aos elementos presentes na solução corrente. Adicionalmente, neste mesmo trabalho foram apresentadas definições formais para o cálculo das distâncias entre as tarefas, utilizadas como pesos das arestas nos métodos de Crama et al. (1994) e foi definida uma função objetivo para o PCV que considera o KTNS como um de seus componentes. Das heurísticas citadas, a que apresentou melhores resultados foi a versão do *GENIUS* que incorpora o KTNS.

Os primeiros a utilizarem dados reais oriundos das indústrias foram Shirazi and Frizelle (2001), provando que os métodos existentes na literatura eram melhores que os utilizados nas indústrias. Observou-se que as instâncias reais possuíam uma capacidade  $C$  elevada em relação as instâncias da literatura, de forma que as instâncias reais possuíam solução mais fácil do que as da literatura.

Diferentes problemas de sequenciamento de produção foram relacionados em Linhares and Yanasse (2002), como o GMLP, o Problema de Pilhas Abertas (*Minimization of Open Stacks Problem* – MOSP), o Problema de Minimização de Descontinuidades (*Minimization of Discontinuities Problem* – MDP) e Problema de Minimização de Espalhamento de Ordens (*Minimization of Order Spread Problem* – MORP). Entre os resultados teóricos deste trabalho, destaca-se a prova de que MTSP e MOSP, apesar de relacionados, não são equivalentes entre si.

Al-Fawzan and Al-Sultan (2003) propuseram uma busca tabu, utilizando diferentes estratégias e métodos probabilísticos para reduzir o espaço de busca. Adicionalmente, foram utilizados dois métodos para geração de soluções aleatórias: o *Random Swapping*, que realiza trocas entre pares de tarefas, e o *Random Block Insertion*, que realiza trocas entre pares de blocos de tarefas consecutivas. Comparados com a Busca Tabu original, a heurística mista, ou seja, a que utilizou estratégias probabilísticas e de oscilação, foi a que apresentou melhores resultados.

Os métodos *branch-and-cut* e *branch-and-bound* foram aplicados à solução do MTSP por Laporte et al. (2004). Novamente, a formulação baseou-se no PCV. O *branch-and-bound* foi capaz de solucionar instâncias com até 25 tarefas e 25 ferramentas em tempo computacional aceitável. Adicionalmente, foi proposta uma nova função objetivo também baseada no PCV. Um novo *branch-and-bound* foi a base para a implementação do *beam search* de Zhou et al. (2005), que reduziu o espaço de busca ao limitar o número de nós na árvore de busca. Os resultados apresentados foram melhores que os resultados reportados por Bard (1988).

Uma extensão do MTSP que considera ferramentas de tamanho não uniforme, foi abordada por Tzur and Altman (2004). Neste trabalho, foi proposto o método *Aladdin*, baseado no melhor método existente até então para resolver o MTSP original, o GENIUS, utilizado anteriormente por Hertz et al. (1998). Comparou com os outros métodos existentes para o MTSP e implementados por ele. Nessas implementações utilizou-se o método *Keep Smaller Tools Needed Soonest* (KSTNS), que é quem determina o número de troca de ferramentas para a extensão do MTSP abordada. Esta mesma extensão do MTSP foi considerada por Crama et al. (2007). Neste trabalho, concluiu-se que, quando a capacidade de ferramentas na máquina é fixa, o problema pode ser resolvido em tempo determinístico polinomial, utilizando um método proposto no mesmo trabalho. Contudo, este método apresenta um alto custo exponencial na resolução do problema.

Um algoritmo memético foi utilizado por Amaya et al. (2008), cuja implementação utilizou uma variação de algoritmo genético e a busca local *hill climbing*. Além disso, uma implementação de algoritmo memético que utiliza a busca local e o algoritmo genético. Os resultados foram comparados com o *Beam Search* de Zhou et al. (2005) e foram melhores, principalmente os obtidos pelo algoritmo memético.

Outro *branch-and-bound* foi proposto por Yanasse et al. (2009), juntamente com um método para determinar limitantes inferiores para o valor das soluções. Os resultados foram comparados com os resultados obtidos por Laporte et al. (2004). O *branch-and-bound* foi capaz de resolver instâncias não resolvidas anteriormente pelo método comparado, entretanto, não foi capaz de resolver outras instâncias, como alguns casos com 25 tarefas. Baseados neste *branch-and-bound*, Senne and Yanasse (2009) empregaram o método *beam search*, que seleciona as regiões promissoras, com o objetivo de determinar limitantes superiores para acelerar os métodos já existentes, ou seja, reduzir o espaço de busca para os algoritmos. Os resultados obtidos foram comparados com os obtidos por Laporte et al. (2004). O método foi capaz de obter 92% de soluções entre ótimas e melhores soluções existentes.

A pesquisa em algoritmos meméticos teve sequência por Amaya et al. (2012). Neste trabalho, o algoritmo memético foi relacionado com técnicas utilizadas em problemas de otimização em redes. Os resultados foram comparados com os de Amaya et al. (2008), superando-os.

Uma nova modelagem em grafos, porém, não relacionada ao PCV, foi apresentada por Cha-

ves et al. (2012). A solução do MTSP foi dividida em duas fases: construtiva e refinamento. Na fase construtiva gera-se uma solução viável e, posteriormente, a fase de refinamento é responsável por encontrar o máximo local para a solução inicial encontrada na primeira fase, através de uma Busca Local Iterada (*Iterated Local Search* – ILS). As soluções encontradas foram utilizadas como limitantes superiores para o algoritmo *branch-and-bound*, proposto anteriormente por Yanasse et al. (2009). O algoritmo apresentou uma diminuição de 74% dos nós gerados na árvore de busca e de 76% no tempo computacional.

Novamente, Amaya et al. (2013) propuseram diferentes formulações para algoritmos meméticos e genéticos. Desta vez, os algoritmos foram combinados com entropia cruzada e diferentes buscas locais. Os resultados foram comparados com os reportados em Amaya et al. (2012), superando-os. Foi também relatado que os algoritmos que utilizam a entropia cruzada são melhores que os que não utilizam, não havendo um método dominante.

Outra extensão do MTSP foi estudado por Raduly-Baka and Nevalainen (2015), considerando posições fixas para as ferramentas na máquina, isto é, uma ferramenta não pode ser instalada em qualquer posição livre não máquina flexível. Foi provado que esse problema é mais complexo que o MTSP original, sendo NP-difícil no sentido forte. No caso em que o número de módulos alimentadores e o número de módulos alimentadores forem fixos, pode-se encontrar a solução em tempo determinístico polinomial, porém, o expoente polinomial é muito alto.

Uma política alternativa ao KTNS foi recentemente proposta por Adjashvili et al. (2015). Embora o novo método exija tempo determinístico polinomial para determinar um plano de troca de ferramentas, assim como o KTNS, não é realizada comparação entre os dois.

Chaves et al. (2016) apresentaram um método baseado em *Clustering Search* e Algoritmo Genético de Chaves Aleatórias Viciadas (*Biased Random Key Genetic Algorithm* – BRKGA). A metaheurística *Clustering Search* identifica regiões promissoras dentro do espaço de busca e realiza a busca local pelo método Descida em Vizinhança Variável (*Variable Neighborhood Descent* – VND) nessas regiões. A metaheurística BRKGA foi utilizada para gerar soluções dentro da região do espaço de busca determinada pelo *Clustering Search*. Os resultados obtidos por Chaves et al. (2012) nas instâncias propostas por Yanasse et al. (2009) e Crama et al. (1994) foram comparados nos experimentos computacionais conduzidos. O novo método foi capaz de igualar ou superar todos os resultados anteriores para estas instâncias, representando o estado da arte em metaheurísticas relacionadas ao MTSP atualmente.

# Capítulo 4

## Materiais e Métodos

Neste capítulo é abordado, em subseções, a modelagem adotada para o Problema de Minimização de Trocas de Ferramentas e como este se relaciona com o clássico Problema do Caixeiro Viajante. Também é apresentado brevemente o método exato que representa o estado da arte para solução do Problema do Caixeiro Viajante.

### 4.1 Modelagem do Problema

Conforme apresentado no Capítulo 3, diferentes autores relacionaram o MTSP com um problema clássico da literatura, o *Problema do Caixeiro Viajante* (PCV), em inglês *Traveling Salesman Problem*.

Em seu livro (Cook, 2012), William J. Cook, uma referência sobre o estudo do PCV, relata que a origem do problema não possui datamento correto, e faz uma análise das origens históricas do mesmo. Um relato indica que em 1925 uma empresa dos EUA, *Page Seed Company*, deixou para seu funcionário, *Mr. Cleveland*, uma lista de cidades para visitar em busca de negócios. *Mr. Cleveland* e a empresa fizeram observações que deixaram claro o objetivo de minimizar o tempo gasto na rota. Outro relato cita que em 1832, um livro alemão fez a descrição do PCV, em que é indicado que a idéia principal do problema é sempre visitar o máximo de cidades possíveis sem ter que visitá-las duas vezes. Um terceiro relato indica que os primeiros estudos matemáticos relacionados ao PCV foram datados de 1930, em Harvard e Viena, apresentando o problema para a comunidade matemática.

O PCV é um problema combinatório que em sua versão de otimização consiste em, dadas uma lista de  $n$  cidades e as distâncias entre elas, determinar uma rota que comece e termine em uma mesma cidade e passe pelas demais cidades exatamente uma vez, com o menor custo possível. A Figura 4.1 demonstra o problema, através de um grafo, no qual os vértices representam as cidades e as arestas representam as ligações entre duas cidades, cujas distâncias são representadas pelos pesos das arestas.

Computacionalmente, uma instância do PCV pode ser modelada por uma matriz  $W$   $n \times n$ , exemplificada na Tabela 4.1. Na matriz, as linhas e colunas representam cada uma das  $n$  cidades do problema, e cada elemento  $w_{ij}$  da tabela, indica a distância entre as cidades  $i$  e  $j$  ( $i, j = 1 \dots n$ ). Note que  $w_{ij} = w_{ji}$  no caso simétrico e quando  $i = j$ ,  $w_{ij}$  possui valor nulo.

Dado o exemplo da Tabela 4.1, uma solução para o exemplo de instância referida é construída da seguinte maneira: seleciona-se uma cidade qualquer, e a partir dela seleciona-se as demais, de forma a construir um ciclo sem repetições de cidades. A primeira cidade escolhida é a que termina o ciclo. Por exemplo, uma solução  $S = [0, 2, 1, 3, 0]$  demonstra que o ciclo começa e termina na cidade 0, visitando as cidades 2, 1 e 3, nesta ordem, durante o percurso. O

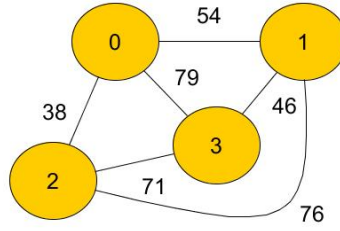


Figura 4.1: Representação de uma instância do PCV.

	0	1	2	3
0	-	54	38	79
1	54	-	76	46
2	38	76	-	71
3	79	46	71	-

Tabela 4.1: Matriz  $W$ .

custo total da solução  $S$  é 239, calculada pela soma das distâncias entre as cidades, de acordo com a Tabela 4.1, apresentadas a seguir:

1. Distância da cidade 0 para 2: 38;
2. Distância da cidade 2 para 1: 76 (acumulado, 114);
3. Distância da cidade 1 para 3: 46 (acumulado, 160); e
4. Distância da cidade 3 para 0: 79 (acumulado, 239).

O objetivo do PCV é determinar a rota que possua o menor custo, ou seja, a solução ótima é aquela que minimiza a distância total percorrida. Para a instância referida na Tabela 4.1, a solução ótima é dado pela sequência  $S = [0, 1, 3, 2, 0]$ , que possui custo total 209.

O PCV, em sua versão de otimização possui complexidade  $\mathcal{NP}$ -Difícil, significando que não se conhece algoritmo eficiente para sua solução. Embora haja diversas heurísticas clássicas para sua solução, estas heurísticas não possuem bom desempenho à medida em que o número de cidades aumenta, tendendo ao aumento rápido do tempo de execução. Da mesma forma, métodos exatos exigem muito tempo para a solução exata, tornando-se impraticáveis.

Novamente, Cook (2012) destaca a evolução dos algoritmos criados para solução do PCV. O primeiro, que permaneceu como estado da arte por 17 anos, resolveu uma instância do problema contendo 49 cidades. Atualmente, o estado da arte é o resolvidor *Concorde*<sup>1</sup>, utilizado para resolver o PCV entre outros problemas, como o roteamento de veículos, o mapeamento genético e outros. Em se tratando do PCV, este resolvidor foi utilizado para resolver problemas existentes na *TSPLIB*<sup>2</sup>, uma biblioteca específica de casos de testes para o PCV. Em seus resultados, se destaca a solução ótima de todos os 110 casos de teste disponíveis, incluindo um caso de teste com 85.900 cidades, assim sendo o maior número de cidades resolvidas atualmente.

<sup>1</sup><http://www.math.uwaterloo.ca/tsp/concorde>

<sup>2</sup><http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>

## 4.2 O Resolvedor Concorde

O resolvedor *Concorde* é um código desenvolvido para solução do PCV disponível também para vários problemas de otimização. O código pode ser utilizado gratuitamente para fins acadêmicos, havendo também um servidor que permite a resolução de instâncias do problema *online*.

Codificado em ANSI C, apesar de contar com métodos heurísticos em seu código com mais de 700 funções, o Concorde é um método exato. A exatidão do resolvedor é devida à utilização do método de plano de cortes, que por sua vez resolve os problemas utilizando técnicas de relaxação linear. Verifica-se também o uso de técnicas para o cálculo de limitantes inferiores: Triangulação de *Delaunay*, Árvore Geradora Mínima e vários geradores de vizinhos mais próximos. Aliadas a esses métodos, são disponibilizadas também cinco heurísticas geradoras de soluções alternativas para o PCV: gulosa (não especificada), *Boruvka* (Boruvka, 1926), *Quick Boruvka* (Applegate et al., 2003), *Nearest Neighbor*, *Chained Lin-Kernighan* (Kernighan and Lin, 1970) e solução aleatória.

Adicionalmente, o resolvedor possui suporte para programação em ambientes paralelos de memória compartilhada e uma interface gráfica opcional. Estas funções permitem ao usuário a criar códigos especializados para problemas que se relacionam com o PCV, como por exemplo o MTSP e problemas de otimização em redes. Também há suporte para os resolvedores para programação linear QSOPT<sup>3</sup> e CPLEX<sup>4</sup>.

Apesar de o resolvedor apresentar resultados expressivos para alguns problemas de otimização, a sua documentação está incompleta, dificultando o entendimento e o uso do mesmo.

## 4.3 O Problema de Minimização de Trocas de Ferramentas e o Problema do Caixeiro Viajante

Conforme mencionado no Capítulo 3, o MTSP pode ser modelado como o PCV, ao relacionar o cálculo do número de trocas de ferramentas do primeiro problema com cálculo da distância entre as cidades do segundo problema, em que a minimização dos valores calculados é abordada na função objetivo de ambos os problemas. Nesta seção, revisitamos as diferentes formas de cálculo de distâncias propostas na literatura ao modelar o MTSP como o PCV.

Hertz et al. (1998) reportaram cinco formas para realizar o cálculo das distâncias  $d_{ij}$  ( $i, j \in T$ ) dada uma instância do MTSP. As três primeiras são conhecidas da literatura e outras duas novas são propostas. Nas equações a seguir,  $C$  é a capacidade máxima da máquina e  $F_i$  ( $F_i \in F$ ) é conjunto de ferramentas necessárias a processar a tarefa  $i$  ( $i \in T$ ).

A primeira métrica relatada, apresentada na Equação 4.1, relaciona a capacidade máxima da máquina e o número de ferramentas em comum entre duas tarefas. Deste modo, determina-se quantas ferramentas permanecerão na máquina e a capacidade restante da máquina é utilizada como um limitante superior para o número de trocas.

$$d_{ij} = C - |F_i \cap F_j| \quad (4.1)$$

Para a referida Equação 4.1, Hertz et al. (1998) citaram que o MTSP se reduz ao PCV caso todas as tarefas  $T_i$  ( $T_i \in T$ ) necessitem de exatamente  $C$  ferramentas para serem processadas. Além disso, reportaram que o caso geral apresenta  $|T_i| < C$  ( $T_i \in T$ ) e ainda assim tem-se algoritmos para o PCV que resultam em bons resultados heurísticos para o problema.

<sup>3</sup><http://www.math.uwaterloo.ca/~bico/qsopt>

<sup>4</sup><http://www.ibm.com/software/commerce/optimization/cplex-optimizer/>

A Equação 4.2 considera como limitante superior para o número de troca de ferramentas entre duas tarefas a diferença simétrica entre os conjuntos de ferramentas necessárias para processar duas tarefas diferentes (i.e., a união dos conjuntos subtraída da interseção entre os mesmos).

$$d_{ij} = |F_i \cup F_j| - |F_i \cap F_j| \quad (4.2)$$

Uma observação válida sobre as Equações 4.1 e 4.2 é que quando  $|F_i| = C$  as equações se tornam equivalentes.

Uma outra métrica foi proposta anteriormente por Tang and Denardo (1988) para cálculo de limitantes inferiores para o número de trocas de ferramentas, apresentada na Equação 4.3. Esta métrica considera o número de ferramentas utilizadas por duas tarefas que excede a capacidade de máquina.

$$d_{ij} = |F_i \cup F_j| - C \quad (4.3)$$

Entretanto, a métrica apresentada na Equação 4.3 pode produzir valores negativos. A terceira métrica enumerada por Hertz et al. (1998) é uma adaptação desta métrica para evitar valores negativos que foi proposta originalmente por Crama et al. (1994). A Equação 4.4 apresenta esta métrica.

$$d_{ij} = \max\{0, |F_i \cup F_j| - C\} \quad (4.4)$$

As três primeiras métricas (Equações 4.1, 4.2 e 4.4) levam em consideração somente duas tarefas subsequentes  $i$  e  $j$ , sem considerar os conjuntos de ferramentas carregadas na máquina antes do processamento da tarefa  $i$  e depois do processamento da tarefa  $j$ . Levando em consideração esta análise, Hertz et al. (1998) propôs duas novas métricas.

A métrica expressada pela Equação 4.5, aperfeiçoa a Equação 4.4. É adotado o critério de frequência de utilização para a permanência das ferramentas na máquina, carregadas anteriormente. Sendo assim, subtrai-se uma quantidade menor que  $C$  de ferramentas caso as ferramentas requeridas pelas tarefas  $i$  ou por  $j$  não forem necessárias antes de  $i$  ou depois de  $j$ . Caso as ferramentas requeridas por  $i$  ou por  $j$  forem requeridas antes de  $i$  ou depois de  $j$ , o número de ferramentas subtraídas será maior. Para isso, computa-se a função  $\lambda_k(ij)$ , resultando no número de tarefas, excluindo-se as tarefas  $i$  e  $j$ , que requerem da ferramenta  $k \in F_i \cup F_j$ , e a função  $\Lambda(ij) = \sum_{k \in F_i \cup F_j} \lambda_k(ij)$ , que representa o total das frequência de todas as ferramentas utilizadas para processar duas tarefas  $i$  e  $j$ . Além disso, tem-se o parâmetro  $\theta \in [0, 1]$ , assim, a métrica subtrai de  $|F_i \cup F_j|$  uma quantidade entre  $[0, C]$ , sendo esta quantidade alta quando as ferramentas de  $F_i \cup F_j$  são utilizadas com frequência.

$$d_{ij} = \max\left\{0, |F_i \cup F_j| - \left\lceil \theta \frac{\Lambda(ij)}{(n-2) |F_i \cup F_j|} \right\rceil C \right\} \quad (4.5)$$

Nesta mesma lógica, a Equação 4.6 aperfeiçoa a Equação 4.2. O fator  $\frac{c+1}{c}$  é um valor entre  $[1, 2]$  que proporciona um peso maior para o termo  $|F_i \cup F_j|$  caso a capacidade máxima da máquina for pequena, i. e., se mais trocas de ferramentas são prováveis. A segunda parte da equação é no mínimo 1, tornando-se maior se as ferramentas presentes em  $F_i \cup F_j$  forem raramente utilizadas. O parâmetro 0.5 é utilizado para evitar a divisão por 0, quando  $\Lambda(ij) = 0$ .

$$d_{ij} = \left( \left\lceil \frac{c+1}{c} \right\rceil |F_i \cup F_j| - |F_i \cap F_j| \right) \left[ \frac{(n-2) |F_i \cup F_j|}{\max\{\Lambda(ij), 0.5\}} \right] \quad (4.6)$$

Em testes realizados nas heurísticas apresentadas em seu trabalho, Hertz et al. (1998) reportaram que o cálculo da distância realizado pela Equação 4.6 foi a que apresentou melhores resultados.



# Capítulo 5

## Conclusão

Seguindo o planejamento, este relatório parcial relatou um trabalho de iniciação científica elaborado em cinco meses de pesquisa.

Foi realizado uma revisão bibliográfica sucinta (3), abordando os principais trabalhos já realizados para o Problema de Minimização de Troca de Ferramentas, gerando assim um relatório que servirá como base teórica para o desenvolvimento de um novo método para o problema.

Neste relatório, faz-se uma analogia do problema com o clássico Problema do Caixeiro Viajante (em inglês, *TSP*), que possui um resolvedor exato, o Concorde (4). Também relata que o método apresenta suporte para problemas do tipo *TSP-like*, portanto abrange o problema em análise. Posteriormente uma modelagem do problema será implementada para invocar o resolvedor Concorde e assim gerar um resultado satisfatório.

Adicionalmente, gerou-se um relatório técnico sobre o Concorde, que será apresentado a comunidade acadêmica, já que o resolvedor não possui grande documentação para a sua instalação e o seu uso.

# Referências Bibliográficas

- David Adjashvili, Sandro Bosio, and Kevin Zemmer. Minimizing the number of switch instances on a flexible machine in polynomial time. *Operations Research Letters*, 43(3):317–322, 2015.
- M A Al-Fawzan and K S Al-Sultan. A tabu search based algorithm for minimizing the number of tool switches on a flexible machine. *Computers & industrial engineering*, 44(1):35–47, 2003.
- Jhon Edgar Amaya, Carlos Cotta, and Antonio J Fernández. A memetic algorithm for the tool switching problem. In *Hybrid metaheuristics*, pages 190–202. Springer, 2008.
- Jhon Edgar Amaya, Carlos Cotta, and Antonio J Fernández-Leiva. Solving the tool switching problem with memetic algorithms. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 26(02):221–235, 2012.
- Jhon Edgar Amaya, Carlos Cotta, and Antonio J Fernández-Leiva. Cross entropy-based memetic algorithms: An application study over the tool switching problem. *International Journal of Computational Intelligence Systems*, 6(3):559–584, 2013.
- David Applegate, William Cook, and André Rohe. Chained lin-kernighan for large traveling salesman problems. *INFORMS Journal on Computing*, 15(1):82–92, 2003.
- Jonathan F Bard. A heuristic for minimizing the number of tool switches on a flexible machine. *IIE transactions*, 20(4):382–391, 1988.
- O Boruvka. O jistem problemu minimaalnim. moravske prirodovedecke spolecnosti 3, 37-58, 1926.
- AA Chaves, LAN Lorena, ELF Senne, and MGC Resende. Hybrid method with cs and brkga applied to the minimization of tool switches problem. *Computers & Operations Research*, 67:174–183, 2016.
- Antônio Augusto Chaves, Edson Luiz França Senne, and Horacio Hideki Yanasse. A new heuristic for the minimization of tool switches problem. *Gestão & Produção*, 19(1):17–30, 2012.
- W. Cook. *In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation*. Princeton University Press, 2012. ISBN 9780691152707. URL [https://books.google.com.br/books?id=S3bxbr\\\_ -qhYC](https://books.google.com.br/books?id=S3bxbr\_ -qhYC).
- Yves Crama, Antoon W. J. Kolen, Alwin G. Oerlemans, and Frits C. R. Spieksma. Minimizing the number of tool switches on a flexible machine. *International Journal of Flexible Manufacturing Systems*, 6(1):33–54, 1994. ISSN 1572-9370. doi: 10.1007/BF01324874. URL <http://dx.doi.org/10.1007/BF01324874>.

- Yves Crama, Linda S Moonen, Frits CR Spieksma, and Ellen Talloen. The tool switching problem revisited. *European Journal of Operational Research*, 182(2):952–957, 2007.
- Michel Gendreau, Alain Hertz, and Gilbert Laporte. New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research*, 40(6):1086–1094, 1992.
- B Golden and W Stewart. Empirical analysis of heuristics in the travelling salesman problem, e. lawer, j. lenstra, a. rinnooy and d. shmoys, 1985.
- Alain Hertz, Gilbert Laporte, Michel Mittaz, and Kathryn E Stecke. Heuristics for minimizing tool switches when scheduling part types on a flexible machine. *IIE transactions*, 30(8): 689–694, 1998.
- Brian W Kernighan and Shen Lin. An efficient heuristic procedure for partitioning graphs. *Bell system technical journal*, 49(2):291–307, 1970.
- Gilbert Laporte, Juan Jose Salazar-Gonzalez, and Frederic Semet. Exact algorithms for the job sequencing and tool switching problem. *IIE Transactions*, 36(1):37–45, 2004.
- Alexandre Linhares and Horacio Hideki Yanasse. Connections between cutting-pattern sequencing, {VLSI} design, and flexible machines. *Computers & Operations Research*, 29(12):1759 – 1772, 2002. ISSN 0305-0548. doi: [http://dx.doi.org/10.1016/S0305-0548\(01\)00054-5](http://dx.doi.org/10.1016/S0305-0548(01)00054-5). URL <http://www.sciencedirect.com/science/article/pii/S0305054801000545>.
- Csaba Raduly-Baka and Olli S Nevalainen. The modular tool switching problem. *European Journal of Operational Research*, 242(1):100–106, 2015.
- Edson Luiz França Senne and Horacio Hideki Yanasse. Beam search algorithms for minimizing tool switches on a flexible manufacturing system. In *XI WSEAS International Conference on Mathematical and Computational Methods in Science and Engineering, MACMESE*, volume 9, pages 68–72, 2009.
- Rabia Shirazi and GDM Frizelle. Minimizing the number of tool switches on a flexible machine: an empirical study. *International Journal of Production Research*, 39(15):3547–3560, 2001.
- Christopher S Tang and Eric V Denardo. Models arising from a flexible manufacturing machine, part i: minimization of the number of tool switches. *Operations research*, 36(5):767–777, 1988.
- Michal Tzur and Avri Altman. Minimization of tool switches for a flexible manufacturing machine with slot assignment of different tool sizes. *IIE Transactions*, 36(2):95–110, 2004.
- Horacio Hideki Yanasse, Rita de Cássia Meneses Rodrigues, and Edson Luiz França Senne. Um algoritmo enumerativo baseado em ordenamento parcial para resolução do problema de minimização de trocas de ferramentas. *Gestão and Produção*, 13(3), 2009.
- Bing-Hai Zhou, Li-Feng Xi, and Yong-Shang Cao. A beam-search-based algorithm for the tool switching problem on a flexible machine. *The International Journal of Advanced Manufacturing Technology*, 25(9-10):876–882, 2005.