

Uma Avaliação Precisa da Modelagem do Problema de Minimização de Troca de Ferramentas como o Problema do Caixeiro Viajante

Túlio Neme de Azevedo

Departamento de Ciência da Computação, Universidade Federal de Ouro Preto
Campus Morro do Cruzeiro, Ouro Preto, Minas Gerais, 35400-000, Brasil.
tulioneme10@gmail.com

Marco Antonio Moreira de Carvalho

Departamento de Ciência da Computação, Universidade Federal de Ouro Preto
Campus Morro do Cruzeiro, Ouro Preto, Minas Gerais, 35400-000, Brasil.
mamc@iceb.ufop.br

RESUMO

Um problema comum encontrado nas linhas de produção industriais caracteriza-se por fabricar um conjunto de diferentes produtos, cada um processado utilizando um conjunto de ferramentas específico em uma máquina flexível. Tal máquina possui uma capacidade fixa para comportar ferramentas e, para processar produtos consecutivamente, trocas de ferramentas são necessárias, implicando em pausas na produção. O Problema de Minimização de Troca de Ferramentas tem como objetivo obter o sequenciamento dos produtos que minimize o número de troca de ferramentas. Este trabalho revisa uma modelagem reportada com frequência na literatura, envolvendo o Problema do Caixeiro Viajante (PCV). Observa-se que somente métodos heurísticos foram utilizados anteriormente para solução do PCV, contribuindo para a degradação da qualidade da solução. Para avaliar precisamente esta modelagem, utilizou-se o resolvidor exato Concorde. Experimentos computacionais indicaram baixa qualidade das soluções, concluindo que a modelagem não é adequada para o problema analisado, considerando as métricas presentes na literatura.

PALAVRAS CHAVE. Minimização de Troca de Ferramentas, Concorde, Caixeiro Viajante.

Tópicos : IND, OC.

ABSTRACT

A common problem found in industrial production lines is characterized by processing a set of different products, each requiring a specific set of tools, in a flexible machine. Such machines have a fixed capacity to store tools and, in order to process all products tool changes are required, implying pauses in production line. The Minimization of Tool Switches Problem aims to obtain the a sequencing of products that minimizes the number of tool switches. This work revises one modeling frequently reported in the literature, involving the Traveling Salesman Problem (TSP). We observe that heuristic methods were previously used to the TSP solution, contributing to the degradation of the quality of solutions. To precisely evaluate this modeling, we use the Concorde exact TSP solver. Computational experiments indicated low quality of the solutions, concluding that the modeling is not adequate for the analyzed problem, considering the metrics present in the literature.

KEYWORDS. Minimization of Tool Switches. Concorde. Traveling Salesman.

Tópicos: IND, OC

1. Introdução

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 simultaneamente, por exemplo, lâminas de corte e brocas, entre outras. Diferentes conjuntos de ferramentas são exigidos para produzir diferentes tipos de produtos e cada produto exige que o respectivo conjunto de ferramentas esteja instalado na máquina flexível 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, durante a fabricação dos produtos, trocas de ferramentas serão necessárias para que esta capacidade não seja ultrapassada e o produto subsequente seja processado. Assim, ao trocar as ferramentas da máquina, a linha de produção é interrompida, gerando ociosidade. O *Problema de Minimização de Troca de Ferramentas* (ou *Minimization of Tool Switches Problem*, MTSP) consiste em determinar a sequência de fabricação dos produtos a cada estágio nas linhas de produção, em que a fabricação de um produto é considerada uma *tarefa* a ser realizada. 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-se mais eficiente.

O planejamento de produção em sistemas de manufatura flexível se divide em duas partes: o problema de carregamento de ferramentas e o problema de sequenciamento de tarefas, como definido por Tang e Denardo (1988). O problema de carregamento de ferramentas consiste em, dada uma sequência fixa de tarefas, determinar o menor número de trocas de ferramentas para processá-la. Também definido por Tang e Denardo (1988), o método *Keep Tool Needed Soonest* (KTNS) soluciona esse problema em tempo determinístico polinomial. Quando trocas de ferramentas são necessárias, a política KTNS garante que as ferramentas que serão necessárias mais brevemente pelas tarefas ainda não processadas sejam mantidas na máquina. Já o problema de sequenciamento de tarefas é \mathcal{NP} -Difícil para $C \geq 2$ (Crama et al., 1994), significando que não se conhece algoritmo eficiente para sua solução.

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 $F_i \in F$ necessárias para realizar a tarefa $i \in T$, é necessário determinar uma permutação ϕ 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 (a) apresenta os dados de um cenário relacionado ao MTSP em que tem-se $n = 5$, $m = 5$, $C = 3$.

Tabela 1: Exemplo de instância MTSP (a) e matriz R (b).

Tarefas	Ferramentas	Ferramentas/Tarefas	1	2	3	4	5
1	2, 3, 5	1	0	1	1	1	0
2	1, 3	2	1	0	0	1	1
3	1, 4, 5	3	1	1	0	0	1
4	1, 2	4	0	0	1	0	1
5	2, 3, 4	5	1	0	1	0	0

(a)

(b)

Na referida Tabela 1(a), a primeira coluna apresenta as tarefas a serem processadas pela máquina flexível, enumeradas de 1 a 5. A segunda coluna apresenta as ferramentas necessárias para que cada tarefa seja processada. Computacionalmente, esta instância pode ser modelada por uma

matriz binária R , como apresentado na Tabela 1(b), na qual as linhas representam as ferramentas e as colunas representam as tarefas. Os elementos r_{ij} da matriz R são preenchidos da seguinte maneira: $r_{ij} = 1$ caso a ferramenta $i \in F$ seja necessária para processar a tarefa $j \in T$ e $r_{ij} = 0$ caso contrário.

Modelado o problema, o sequenciamento de tarefas é dado por uma permutação ϕ das colunas de R , por exemplo, $\phi = [3, 4, 2, 1, 5]$. Uma representação do plano de troca de ferramentas, dada a solução ϕ , é feita pela matriz binária R^ϕ , em que configura-se uma troca de ferramenta quando $R^\phi_{i,j-1} = 0$ e $R^\phi_{i,j} = 1$, ou seja, a ausência da ferramenta i durante o processamento da tarefa $j - 1$ e a inserção da mesma no processamento da tarefa j . Desta forma, o total de inversões em R^ϕ indica o número de troca de ferramentas.

A Tabela 2 indica a solução ótima R^ϕ para a instância apresentada pela Tabela 1(b), cujas colunas são as colunas de R na ordem estabelecida por ϕ e uma coluna adicional 0 com todos os elementos nulos, para efeito do cálculo das inversões. Na referida Tabela 2, as ferramentas inseridas a cada estágio na máquina estão sublinhadas e as ferramentas que permanecem na máquina, mesmo sem serem utilizadas, estão indicadas em negrito. Por exemplo, após o processamento da tarefa 3, a ferramenta 2 foi inserida na máquina, portanto sublinhada, e a ferramenta 5 foi mantida na máquina mesmo sem ser utilizada pela tarefa seguinte, portanto em negrito.

Tabela 2: Matriz R^ϕ .

Ferramentas/ ϕ	0	3	4	2	1	5
1	0	<u>1</u>	1	1	0	0
2	0	0	<u>1</u>	1	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>	1	0	<u>1</u>	0

O número de troca de ferramentas do exemplo referido pela Tabela 2 é 7, considerando a inserção das três ferramentas iniciais (1, 4 e 5), uma troca entre as tarefas 3 e 4 (ferramenta 4 pela 2), uma troca entre as tarefas 4 e 2 (ferramenta 5 pela 3), uma troca entre as tarefas 2 e 1 (ferramenta 1 pela 5) e uma troca entre as tarefas 1 e 5 (ferramenta 5 pela 4).

Este trabalho tem como objetivo analisar com exatidão a modelagem mais comum para solução do MTSP reportada na literatura. Em trabalhos anteriores, após a modelagem utiliza-se métodos heurísticos para solução do problema, o que impede a análise precisa da modelagem adotada. Propõe-se, portanto, a solução exata de tal modelagem, de maneira a evidenciar sua qualidade inequivocamente.

O restante deste trabalho está organizado da seguinte forma: a Seção 2 apresenta uma revisão geral da literatura sobre o MTSP, seguida da descrição da modelagem analisada na Seção 3. Experimentos computacionais envolvendo os modelos analisados são apresentados na Seção 5 e por fim, a Seção 6 contém as conclusões deste trabalho e também relaciona possibilidades de trabalhos futuros.

2. Revisão da Literatura

Os primeiros autores a abordarem o MTSP foram Tang e Denardo (1988), sugerindo a modelagem do problema como o Problema do Caminho Hamiltoniano Mínimo em um grafo completo em que os vértices representam as tarefas e cujos pesos das arestas foi definido como um limitante inferior para o número de troca de ferramentas entre duas tarefas específicas. Contudo, 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 processadas. O critério para escolha de quais ferramentas devem ser retiradas da máquina a cada troca é baseado na proximidade do uso de cada ferramenta, de forma que ferramentas necessárias para processar as próximas tarefas

sejam mantidas sempre na máquina. 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.

Crama et al. (1994) modelaram o MTSP como o Problema do Caixeiro Viajante (PCV), conforme descrito em detalhes na Seção 3. Foram utilizadas heurísticas específicas para resolução da do PCV, como *nearest neighbor*, *farthest insertion* e *shortest edge* aliadas a estratégias de busca local, como a *global-2-opt*. Utilizando a mesma modelagem, Hertz et al. (1998) também utilizaram heurísticas próprias do PCV, como *farthest insertion*, GENI e GENIUS.

Novamente, a modelagem pelo PCV foi considerada em Privault e Finke (2000). Foi proposta uma adaptação do método *Partitioning Algorithm*, cujos resultados foram comparados aos de métodos específicos para o PCV *Farthest Insertion*, *2-opt* e *Shortest Edge*. O método proposto apresentou boas soluções, porém o tempo computacional foi muito maior que os comparados. Djellab et al. (2000) propuseram uma modelagem do MTSP através de hipergrafos, associada ao método *Iterative Best Insertion*. Nos experimentos realizados, foi reportada a comparação com os métodos de Crama et al. (1994), sendo que o método proposto obteve resultados competitivos.

Dados reais oriundos das indústrias foram utilizados por Shirazi e 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.

Uma modelagem do MTSP como o Problema de Fluxo de Custo Mínimo em Redes foi proposta por Yanasse e Pinto (2002), entretanto, não foram realizados experimentos computacionais. Posteriormente, Yanasse e Lamosa (2005) enunciaram uma proposta de modelagem utilizando o Problema do Caixeiro Viajante Generalizado, implementada por Yanasse e Lamosa (2006), porém, em um artigo teórico.

Al-Fawzan e Al-Sultan (2003) propuseram uma busca tabu, utilizando diferentes estratégias e métodos probabilísticos para reduzir o espaço de busca. 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, juntamente com a heurística GENIUS, para construir a solução do MTSP por Laporte et al. (2004). Neste trabalho foi proposta uma nova função objetivo, no intuito de melhorar a relaxação linear de um modelo baseado no PCV. Essa nova proposta introduziu restrições oriundos do PCV à formulação do MTSP reportada por Tang e Denardo (1988), resultando assim na nova função objetivo.

A metaheurística *Beam Search* foi aplicada ao MTSP por Zhou et al. (2005) e obteve melhores resultados em relação àqueles obtidos por Bard (1988). Amaya et al. (2008) propôs a utilização de algoritmos meméticos e superou os resultados da metaheurística *Beam Search*. Posteriormente, Amaya et al. (2012) e Amaya et al. (2013) aprimoraram a implementação de algoritmos meméticos, obtendo melhores resultados quando comparado com a versão anterior do método.

Métodos para obtenção de limitantes inferiores e um método *branch-and-bound* foram introduzidos por Yanasse et al. (2009). O *branch-and-bound* foi capaz de resolver instâncias não resolvidas anteriormente pelo método de Laporte et al. (2004), entretanto, não foi capaz de resolver outras instâncias, como alguns casos com 25 tarefas. Baseados neste mesmo *branch-and-bound*, Senne e Yanasse (2009) propuseram uma nova implementação do método *beam search*, capaz de obter 92% de soluções entre ótimas e melhores soluções existentes.

Uma implementação da metaheurística Busca Local Iterada foi apresentada por Chaves et al. (2012). Inicialmente, uma heurística construtiva baseada na representação do MTSP por grafos gera uma solução, posteriormente, refinada pela busca local iterada. As soluções encontradas foram utilizadas como limitantes superiores para o algoritmo *branch-and-bound*, proposto anteriormente por Yanasse et al. (2009), gerando uma redução de 74% no número de nós gerados na árvore de busca e de 76% no tempo computacional.

Três modelos de programação linear inteira específicos para solucionar o MTSP foram implementados por Catanzaro et al. (2015). Foram analisados e modificados os métodos propostos anteriormente por Tang e Denardo (1988) e Laporte et al. (2004). Embora os resultados reportados indiquem que ambos os novos métodos apresentam melhor relaxação linear e resolvam um maior número de instâncias, os autores fazem a ressalva de que, em geral, o valor de *gap* e o alto número de variáveis pode limitar a utilização prática dos modelos.

Recentemente, Chaves et al. (2016) apresentaram um método baseado em *Clustering Search* e Algoritmo Genético de Chaves Aleatórias Viciadas. O novo método foi capaz de igualar ou superar todos os resultados anteriores obtidos por Chaves et al. (2012) para as instâncias propostas por Yanasse et al. (2009) e Crama et al. (1994). O atual estado da arte para o MTSP consiste de uma nova implementação de Busca local Iterada, proposta por Paiva e Carvalho (2016). Foram propostos neste trabalho uma nova representação em grafos para o problema, uma nova heurística construtiva para a geração de soluções iniciais e um novo método de busca local específico para o problema. Os resultados destes métodos foram comparados aos resultados obtidos por Chaves et al. (2016), sendo capazes de igualar ou superar todos os resultados anteriores para essas instâncias.

3. MTSP Modelado como o Problema do Caixeiro Viajante

Uma maneira comum para solucionar problemas de otimização consiste em realizar modelagens de problemas correlatos entre si que possuam métodos de solução de boa qualidade. Desta forma, a solução do problema original é composta da solução gerada pela modelagem para um segundo problema. Conforme descrito na Seção 2, uma da modelagem frequente na literatura sobre o MTSP envolve o *Problema do Caixeiro Viajante* (ou PCV).

O PCV é um clássico problema combinatório, que em sua versão de otimização consiste em, dada uma lista de w cidades e as respectivas distâncias entre cada par de cidades, determinar uma rota com menor custo possível que se inicia e termina em uma mesma cidade, passando por todas as demais cidades exatamente uma vez. Este é um problema modelado em grafos, tal que cada cidade é representada por um vértice e as arestas representam as conexões entre as cidades, cujos pesos são as distâncias. O PCV, em sua versão de otimização é um problema \mathcal{NP} -Difícil, como definido por Laporte (1992).

O MTSP é equivalente ao PCV apenas no caso específico em que cada tarefa requer exatamente C ferramentas, como apontado por Hertz et al. (1998). Embora o MTSP e o PCV não sejam equivalentes entre si no caso geral, uma modelagem comum utilizando ambos os problemas considera um grafo completo em que cada tarefa é representada por um vértice. Os pesos das arestas podem ser definidos de diferentes maneiras, de forma a converter o número de troca de ferramentas entre duas tarefas processadas contiguamente, característica do MTSP, na distância entre cidades, característica do PCV.

A Figura 1 ilustra essa modelagem, na qual cada vértice do grafo em análise representa uma cidade para o PCV e uma tarefa para o MTSP. Já as arestas do grafo indicam as distâncias entre duas cidades para o PCV e o custo em sequenciar duas tarefas para o MTSP. Portanto utiliza-se a modelagem MTSP via PCV para definir a sequência de tarefas e posteriormente aplicar o método KTNS para o cálculo definitivo do número de trocas de ferramentas. Assim, uma dificuldade desta modelagem reside em definir uma métrica adequada para cálculo de distância no PCV que gere a melhor solução para o MTSP.

Além da dificuldade introduzida pela definição dos pesos das arestas, a análise da literatura realizada na Seção 2 indica que somente métodos heurísticos foram considerados para solução dos modelos. Desta forma, as soluções geradas possuem, além da degradação da modelagem por si só, a degradação oriunda da solução por heurísticas, o que dificulta uma avaliação precisa da modelagem.

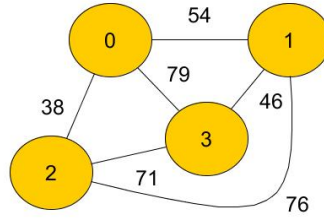


Figura 1: Modelando MTSP como PCV.

3.1. Estimativas do Número de Trocas de Ferramentas

Cinco métricas para transformação do número de trocas de ferramentas entre duas tarefas processadas contiguamente no MTSP em distâncias do PCV são propostas na literatura. A seguir, estas são apresentadas e analisadas. Nas equações, d_{ij} representa a distância entre a tarefa i e j .

A primeira métrica, apresentada na Equação 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. Esta métrica foi adotada por Hertz et al. (1998).

$$d_{ij} = C - |F_i \cap F_j| \quad (1)$$

A Equação 2 apresenta a métrica que 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). Hertz et al. (1998) utilizaram essa métrica em seus modelos.

$$d_{ij} = |F_i \cup F_j| - |F_i \cap F_j| \quad (2)$$

A terceira métrica (Equação 3), proposta originalmente por Crama et al. (1994), baseia-se no cálculo de limitantes inferiores para o número de trocas de ferramentas e evita valores negativos. Este é o cálculo de distância utilizada pela maioria dos autores que solucionaram o MTSP utilizando a modelagem baseada no PCV (Tang e Denardo, 1988; Crama et al., 1994; Hertz et al., 1998; Laporte et al., 2004; Privault e Finke, 2000; Catanzaro et al., 2015).

$$d_{ij} = \max\{0, |F_i \cup F_j| - C\} \quad (3)$$

As três primeiras métricas (Equações 1, 2 e 3) 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, aperfeiçoa a métrica da Equação 3. 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 sejam necessárias antes de i ou depois de j . Caso as ferramentas requeridas por i ou por j sejam requeridas também antes de i ou depois de j , o número de ferramentas subtraídas será maior. Para isso, computa-se a função $\lambda_k(ij)$, que representa o número de tarefas, excluindo-se as tarefas i e j , que requerem cada 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 somatório das frequências de utilização de todas as ferramentas exigidas 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. Esta métrica foi utilizada somente por Hertz et al. (1998).

$$d_{ij} = \max \left\{ 0, |F_i \cup F_j| - \left\lceil \theta \frac{\Lambda(ij)}{(n-2) |F_i \cup F_j|} \right\rceil C \right\} \quad (4)$$

De maneira semelhante, a métrica apresentada na Equação 5 aperfeiçoa a métrica descrita pela Equação 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. O segundo fator 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 $\Lambda(ij) = 0$. Assim como na métrica anterior, somente Hertz et al. (1998) utilizaram essa métrica em seus experimentos.

$$d_{ij} = \left(\left\lceil \frac{c+1}{c} \right\rceil |F_i \cup F_j| - |F_i \cap F_j| \right) \left[\frac{(n-2) |F_i \cup F_j|}{\max\{\Lambda(ij), 0,5\}} \right] \quad (5)$$

O único trabalho da literatura a considerar todas as métricas apresentadas, Hertz et al. (1998) conclui que a métrica que gera melhores soluções para o MTSP é a apresentada na Equação 5. Todas as métricas analisadas serão novamente submetidas a comparações entre si e com relação ao estado da arte na Seção 5.

4. Métodos Propostos

De acordo com as análises realizadas, constatou-se que uma forma de diminuir o nível de degradação das soluções obtidas quando modela-se o MTSP como o PCV é não utilizar métodos heurísticos para solução do PCV nesta situação. A utilização de métodos exatos neste caso, além de gerar as soluções ótimas, permite que a modelagem do MTSP como o PCV seja analisada precisamente.

Atualmente, o estado da arte para solução exata do PCV é representado pelo resolvidor *Concorde*¹. Apesar de contar com métodos heurísticos em seu código com mais de 700 funções, o Concorde é um método exato, o que se deve 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, como Triangulação de *Delaunay*, Árvore Geradora Mínima e várias heurísticas do tipo *vizinho mais próximo*. Aliadas a esses métodos, são disponibilizadas também cinco heurísticas geradoras de soluções alternativas para o PCV: gulosa (não especificada), *Boruvka*, *Quick Boruvka*, *Nearest Neighbor*, *Chained Lin-Kernighan* e solução aleatória. 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 resolvidores de programação linear *QSopt* e *CPLEX*.

Desta forma, propõe-se a modelagem do MTSP como PCV, porém, utilizando-se o resolvidor exato Concorde para solução do PCV, e, posteriormente, a obtenção do número de trocas de ferramentas pela política KTNS. Neste sentido, as cinco diferentes métricas para cálculo de distância mencionadas na Seção 3.1 foram utilizadas para derivar cinco modelos diferentes para a solução do MTSP. São eles: M_1 , M_2 , M_3 , M_4 e M_5 , os quais são associados às Equações 1, 2, 3, 4 e 5, respectivamente.

5. Experimentos Computacionais

Todos os modelos foram implementados utilizando a linguagem C++, além de incorporar as funções do resolvidor Concorde e do resolvidor de programação linear CPLEX, versão 12.6.3. Os experimentos foram realizados em um computador com processador *Intel i3-2350M Quad Core* de 2.30GHz, 4 GB de RAM e sistema operacional Ubuntu 16.04.02. Os resultados obtidos foram comparados com os obtidos pelo atual estado da arte relacionado ao MTSP, a metaheurística *Iterated Local Search* (ILS) proposta por Paiva e Carvalho (2016). Os resultados desse método foram obtidos em um computador com processador *Intel i5 Quad Core* de 3.2GHz e 8 GB de RAM. Além

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

disso, experimentos preliminares foram conduzidos para ajustar o valor do parâmetro θ , de maneira a gerar os melhores resultados quando utilizada a distância M_4 . Entretanto, verificou-se que a variação do valor de θ no intervalo $[0, 1]$ não gerou alteração significativa nas soluções obtidas para as instâncias consideradas.

Três conjuntos de instâncias são consideradas, e os resultados gerados são reportados em dois tipos de tabelas para cada conjunto. O primeiro tipo de tabelas tem o objetivo de apresentar o valor das soluções e o tempo de execução para cada grupo de instâncias. Nelas são apresentados o número de instâncias do conjunto (e), a melhor solução encontrada pelo ILS (S^*), a solução encontrada pelos modelos propostos neste trabalho (S) e o tempo médio de execução de cada modelo em segundos (T). As siglas M_1 , M_2 , M_3 , M_4 e M_5 indicam qual modelo está evidenciado na coluna. O segundo tipo de tabelas tem o objetivo de comparar os modelos propostos entre si e em relação ao estado da arte, apresentando o *gap* (distância percentual em relação à melhor solução existente), calculado como $100 \times (S - S^*)/S^*$ e o *ranking* ordinal, que atribui valores ordinais aos métodos de acordo com a qualidade das soluções associadas, de forma que, quanto melhor a solução, menor o valor de *ranking* – em caso de empate, o mesmo valor ordinal é atribuído aos métodos.

5.1. Instâncias de Yanasse et al. (2009)

Yanasse et al. (2009) propuseram um total de 1.350 instâncias para o MTSP, divididas em 5 grupos (A, B, C, D e E). Entre os grupos, varia-se a capacidade da máquina, o número de tarefas, o número de ferramentas e o número de instâncias. Seguindo a mesma lógica de apresentação de dados sobre os experimentos computacionais de Paiva e Carvalho (2016), por restrição de espaço sumariza-se as soluções obtidas por grupos, conforme apresentado pela Tabela 3. Logo em seguida, na Tabela 4, apresenta-se a análise dessas soluções.

Tabela 3: Resultados obtidos para os grupos A, B, C, D e E.

Grupo	e	ILS		M_1		M_2		M_3		M_4		M_5	
		S^*	T	S	T	S	T	S	T	S	T	S	T
A	340	24,54	0,11	25,10	0,00	25,41	0,00	25,68	0,01	25,23	0,00	26,10	0,00
B	330	25,22	0,18	26,58	0,00	26,23	0,00	26,61	0,01	26,22	0,00	26,79	0,00
C	340	28,92	1,67	31,92	0,12	31,50	0,01	33,55	0,29	31,35	0,02	31,74	0,11
D	260	25,98	6,05	29,18	0,03	29,03	0,02	33,15	0,10	29,06	0,04	29,82	0,03
E	80	16,89	0,51	18,49	0,01	18,43	0,01	20,38	0,03	18,28	0,02	19,64	0,01

Todo os modelos comparados apresentaram tempos de execução menores quando comparados ao ILS. Para as instâncias consideradas mais difíceis do conjunto, as do grupo D, o tempo de execução para os modelos foram inferiores a 0,10 segundo, consideravelmente menores do que os 6,05 segundos exigidos pelo ILS.

Tabela 4: *Ranking* e *gap* obtidos para os grupos A, B, C, D e E.

Grupo	ILS		M_1		M_2		M_3		M_4		M_5	
	<i>gap</i>	<i>ranking</i>	<i>gap</i>	<i>ranking</i>	<i>gap</i>	<i>ranking</i>	<i>gap</i>	<i>ranking</i>	<i>gap</i>	<i>ranking</i>	<i>gap</i>	<i>ranking</i>
A	0,00%	1,00	4,51%	1,85	3,33%	1,63	4,21%	1,78	3,26%	1,61	8,89%	2,32
B	0,00%	1,00	5,58%	2,05	4,18%	1,79	5,43%	2,00	3,83%	1,74	10,99%	2,61
C	0,00%	1,00	14,02%	2,85	12,15%	2,46	17,72%	3,33	12,00%	2,42	17,62%	3,31
D	0,00%	1,00	14,18%	2,65	13,90%	2,57	33,27%	4,22	13,51%	2,52	17,73%	3,13
E	0,00%	1,00	14,02%	2,85	12,15%	2,46	17,72%	3,33	12,00%	2,42	17,62%	3,31

A análise das soluções apresentadas pela Tabela 4 indica claramente que o ILS apresentou os melhores resultados para todos os grupos do conjunto. Embora o *gap* sinalize que para as instâncias menores (grupos A e B) os modelos se aproximaram da solução obtida pelo ILS, o mesmo não ocorre para as instâncias maiores (grupos C, D e E). Com efeito, os modelos sequer se aproximam dos resultados obtidos pelo ILS. O *ranking* evidencia os modelos M_2 e M_4 com melhores médias quando comparados somente os modelos.

Em suma, os resultados obtidos demonstram que, apesar dos baixos tempos de execução, nenhum dos modelos foi capaz de atingir quaisquer dos valores reportados pelo ILS. O modelo M_4 obteve maior destaque entre os modelos propostos, apresentando menores valores de *gap* em 30 dos 50 subgrupos de instâncias. Destaca-se também o mau desempenho dos modelos M_3 e M_5 .

5.2. Instâncias de Crama et al. (1994)

Divididas em 4 grupos (C_1 , C_2 , C_3 e C_4), as 160 instâncias propostas por Crama et al. (1994) são diferenciadas pelo número de tarefas, número de ferramentas e capacidade máxima da máquina flexível. Nas Tabela 5 está apresentado, de forma resumida, as soluções obtidas para cada um dos grupos, e em seguida, a análise dessas soluções é apresentada na Tabela 6

Tabela 5: Resultados obtidos para os grupos C_1 , C_2 , C_3 e C_4 .

Grupo	e	ILS		M_1		M_2		M_3		M_4		M_5	
		S^*	T	S	T	S	T	S	T	S	T	S	T
C_1	40	11,18	0,11	12,15	0,01	11,95	0,01	12,95	0,02	12,03	0,01	12,53	0,01
C_2	40	22,00	0,18	24,05	0,01	24,48	0,01	27,75	0,05	24,55	0,02	25,98	0,02
C_3	40	79,58	175,10	93,95	0,06	93,03	0,04	106,93	0,17	91,85	0,10	95,88	0,08
C_4	40	158,30	1097,28	183,03	0,11	181,03	0,08	208,23	0,22	182,13	0,21	189,75	0,14

Novamente, os tempos de execução dos modelos comparados são baixos, em torno de décimos de segundo, mesmo com a utilização de um resolvidor exato. Por outro lado, pode-se verificar o rápido crescimento do tempo de execução do ILS, particularmente para o grupo C_4 , evidenciando a dificuldade em obter boas soluções para estas instâncias. Entretanto, os modelos propostos apresentaram soluções de baixa qualidade, mesmo para as instâncias de menores dimensões. A qualidade das soluções se deteriorou de maneira mais acentuada na resolução de instâncias dos grupos C_3 e C_4 .

Tabela 6: *Ranking* e *gap* obtidos para os grupos C_1 , C_2 , C_3 e C_4 .

Grupo	ILS		M_1		M_2		M_3		M_4		M_5	
	<i>gap</i>	<i>ranking</i>	<i>gap</i>	<i>ranking</i>	<i>gap</i>	<i>ranking</i>	<i>gap</i>	<i>ranking</i>	<i>gap</i>	<i>ranking</i>	<i>gap</i>	<i>ranking</i>
C_1	0,00%	1,00	8,42%	1,90	6,63%	1,70	15,80%	2,55	7,42%	1,80	11,69%	2,23
C_2	0,00%	1,00	9,17%	2,15	10,22%	2,35	25,06%	3,65	10,97%	2,35	15,40%	2,93
C_3	0,00%	1,00	18,78%	3,40	18,01%	3,03	37,72%	5,33	16,42%	2,58	21,14%	3,98
C_4	0,00%	1,00	16,83%	3,25	15,57%	2,63	34,65%	5,53	16,27%	3,08	21,10%	4,25

Comparando-se os modelos propostos, M_2 se destacou dos demais apresentando melhores soluções em 4 dos 5 dos grupos de instâncias, seguido pelo modelo M_4 . Novamente, a pior performance foi relacionada ao modelo M_3 , embora seguido de perto pelo modelo M_5 .

5.3. Instâncias de Catanzaro et al. (2015)

As 160 instâncias propostas por Catanzaro et al. (2015) foram divididas em 4 grupos (*datA*, *datB*, *datC* e *datD*). Os grupos de instâncias também se diferenciam pelo número de tarefas, número de ferramentas e capacidade da máquina. A Tabela 7 apresenta os resultados e tempos de execução, enquanto os valores de *gap* e *ranking* são apresentados na Tabela 8.

Tabela 7: Resultados obtidos para os grupos *datA*, *datB*, *datC* e *datD*.

Grupo	e	ILS		M_1		M_2		M_3		M_4		M_5	
		S^*	T	S	T	S	T	S	T	S	T	S	T
<i>datA</i>	40	10,85	0,07	11,60	0,01	11,65	0,01	12,88	0,02	11,65	0,01	12,48	0,01
<i>datB</i>	40	21,78	1,01	24,23	0,01	23,55	0,01	27,63	0,05	24,35	0,02	25,18	0,02
<i>datC</i>	40	74,85	174,29	89,48	0,06	88,95	0,04	102,65	0,16	88,30	0,10	92,10	0,07
<i>datD</i>	40	158,33	1052,26	183,28	0,08	181,43	0,11	207,60	0,24	182,58	0,21	189,83	0,17

Os resultados apresentados evidenciam o comportamento dos modelos e do *ILS* em relação ao tempo computacional. Ao passo em que os modelos apresentam pequeno aumento no tempo de execução, novamente em décimos de segundos, o *ILS* apresenta um rápido aumento do tempo de execução de acordo com o aumento das dimensões das instâncias tratadas. Verifica-se novamente que nenhum dos modelos foi capaz de se aproximar das soluções obtidas pelo *ILS*.

Tabela 8: *Ranking* e *gap* obtidos para os grupos *datA*, *datB*, *datC* e *datD*.

Grupo	<i>ILS</i>		M_1		M_2		M_3		M_4		M_5	
	<i>gap</i>	<i>ranking</i>	<i>gap</i>	<i>ranking</i>	<i>gap</i>	<i>ranking</i>	<i>gap</i>	<i>ranking</i>	<i>gap</i>	<i>ranking</i>	<i>gap</i>	<i>ranking</i>
<i>datA</i>	0,00%	1,00	6,72%	1,70	7,23%	1,78	18,96%	2,73	7,24%	1,78	14,53%	2,40
<i>datB</i>	0,00%	1,00	10,62%	2,35	7,71%	1,93	26,83%	3,95	11,11%	2,53	14,62%	2,98
<i>datC</i>	0,00%	1,00	20,23%	3,28	19,71%	3,00	39,64%	5,40	18,74%	2,75	23,48%	3,95
<i>datD</i>	0,00%	1,00	20,23%	3,28	19,71%	3,00	39,64%	5,40	18,74%	2,75	23,48%	3,95

Neste conjunto de instâncias, de maneira geral, apesar do baixo tempo de execução os resultados não são razoavelmente próximos daqueles reportados por heurísticas na literatura recente, assim como nos conjuntos anteriores. Particularmente, nos grupos *datC* e *datD*, considerados como os mais difíceis atualmente, a performance dos modelos propostos foi muito inferior ao atual estado da arte. Comparando-se os modelos entre si, novamente o modelo M_2 obteve os melhores indicadores, seguido pelo modelo M_4 , ao passo que o modelo M_3 obteve os piores indicadores, muito aquém dos demais modelos comparados.

5.4. Discussão

De acordo com os experimentos realizados e suas respectivas análises específicas, é possível realizar uma análise geral sobre a modelagem do Problema de Minimização de Trocas de Ferramentas como o Problema do Caixeiro Viajante. Devido à utilização do resolvedor exato Concorde, foi possível verificar a consistência e qualidade da modelagem em si, dado que os resultados são ótimos para a modelagem utilizada. Isto não foi possível anteriormente na literatura devido à utilização de métodos heurísticos para solução dos modelos, o que por si só poderia gerar degradação da qualidade da solução.

Os dados indicam que as soluções obtidas utilizando a modelagem proposta e as métricas encontradas na literatura possuem baixa qualidade, já que nenhum dos modelos considerados conseguiu se aproximar razoavelmente dos resultados obtidos pelo atual estado da arte, mesmo para instâncias consideradas mais fáceis, ou de dimensões reduzidas. Muito embora os modelos tenham sido capazes de gerar soluções em tempo de execução desprezível, esta característica não é suficiente para que conferir relevância aos mesmos. Outro dado particularmente interessante se refere ao pior desempenho entre os modelos ser atribuído ao modelo M_3 – conforme mencionado na Seção 2, trata-se da estimativa para o número de trocas de ferramentas mais utilizada pelos trabalhos da literatura.

Desta forma, conclui-se que a modelagem do MTSP utilizando PCV foi levada ao limite neste trabalho e que a mesma não é adequada, considerando-se o que já foi proposto neste tema. Todavia, é possível concentrar esforços em novas formas de cálculo de distância que sejam capazes de suprir as deficiências das métricas propostas anteriormente na literatura e analisadas neste trabalho.

6. Conclusões e Trabalhos Futuros

Definido como \mathcal{NP} -Difícil, o Problema de Minimização de Troca de Ferramentas (ou *Minimization of Tool Switches Problem*, MTSP) tem como objetivo obter uma sequência de processamento de tarefas em uma linha de produção que utiliza uma máquina flexível, de maneira a agilizar a produção. Na literatura, uma forma comum de modelar o MTSP baseia-se no clássico Problema do Caixeiro Viajante (PCV). Embora os dois problemas não sejam equivalentes entre si no caso geral, considera-se cada tarefa como uma cidade, tal que existem diferentes formas para

realizar o mapeamento da função objetivo do MTSP para o peso das arestas do PCV. Estas diferentes formas foram aplicadas na modelagem do MTSP pelo PCV, que, via de regra, é resolvido por heurísticas clássicas, fato que influencia negativamente a qualidade das soluções geradas para o MTSP. Neste trabalho, propôs-se revisitar esta tradicional modelagem e também as cinco estimativas para o peso das arestas propostas na literatura sobre o MTSP, além de utilizar o resolvidor exato Concorde para solucionar o PCV. Com base em cada estimativa, derivou-se um modelo para solução do MTSP, o que tornou possível concluir sobre a qualidade da modelagem frequentemente reportada na literatura.

Os experimentos computacionais consideraram três grandes conjuntos de dados, em um total de 1.670 instâncias e os resultados foram comparados aos obtidos pelo método que representa o estado da arte para o problema em questão. Os resultados computacionais reportaram que nenhum modelo foi capaz de se aproximar razoavelmente dos melhores resultados reportados na literatura. As análises dos resultados indicam que dois modelos apresentaram melhores resultados e se destacaram das demais e também, que a estimativa mais comumente utilizada na literatura obteve os piores indicadores em relação aos demais modelos comparados. De maneira geral, foi possível observar que a modelagem do MTSP pelo PCV não é adequada para as instâncias atuais, quando consideramos as métricas existentes para adaptação do MTSP ao PCV.

Os trabalhos futuros se concentrarão em produzir estimativas para os pesos das arestas que apresentem resultados com melhor precisão e melhor desempenho na utilização da modelagem mencionada. Dentre as opções destaca-se a utilização de métodos evolutivos que, utilizadas em conjunto com o resolvidor Concorde, darão origem a heurísticas matemáticas para resolução do MTSP.

7. Agradecimentos

Esta pesquisa foi apoiada pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico, pela Fundação de Apoio à Pesquisa do Estado de Minas Gerais e pela Universidade Federal de Ouro Preto.

Referências

- Al-Fawzan, M. A. e Al-Sultan, K. S. (2003). A tabu search based algorithm for minimizing the number of tool switches on a flexible machine. *Computers & industrial engineering*, 44(1): 35–47.
- Amaya, J. E., Cotta, C., e Fernández, A. J. (2008). A memetic algorithm for the tool switching problem. In *Hybrid metaheuristics*, p. 190–202. Springer.
- Amaya, J. E., Cotta, C., e Fernández-Leiva, A. J. (2012). Solving the tool switching problem with memetic algorithms. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 26(02):221–235.
- Amaya, J. E., Cotta, C., e Fernández-Leiva, A. J. (2013). Cross entropy-based memetic algorithms: An application study over the tool switching problem. *International Journal of Computational Intelligence Systems*, 6(3):559–584.
- Bard, J. F. (1988). A heuristic for minimizing the number of tool switches on a flexible machine. *IIE transactions*, 20(4):382–391.
- Catanzaro, D., Gouveia, L., e Labbé, M. (2015). Improved integer linear programming formulations for the job sequencing and tool switching problem. *European Journal of Operational Research*, 244(3):766–777.
- Chaves, A., Lorena, L., Senne, E., e Resende, M. (2016). Hybrid method with cs and brkga applied to the minimization of tool switches problem. *Computers & Operations Research*, 67:174–183.

- Chaves, A. A., Senne, E. L. F., e Yanasse, H. H. (2012). A new heuristic for the minimization of tool switches problem. *Gestão & Produção*, 19(1):17–30.
- Crama, Y., Kolen, A. W. J., Oerlemans, A. G., e Spieksma, F. C. R. (1994). Minimizing the number of tool switches on a flexible machine. *International Journal of Flexible Manufacturing Systems*, 6(1):33–54. ISSN 1572-9370. URL <http://dx.doi.org/10.1007/BF01324874>.
- Djellab, H., Djellab, K., e Gourgand, M. (2000). A new heuristic based on a hypergraph representation for the tool switching problem. *International Journal of Production Economics*, 64(1): 165–176.
- Hertz, A., Laporte, G., Mittaz, M., e Stecke, K. E. (1998). Heuristics for minimizing tool switches when scheduling part types on a flexible machine. *IIE transactions*, 30(8):689–694.
- Laporte, G. (1992). The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(2):231–247.
- Laporte, G., Salazar-Gonzalez, J. J., e Semet, F. (2004). Exact algorithms for the job sequencing and tool switching problem. *IIE Transactions*, 36(1):37–45.
- Paiva, G. S. e Carvalho, M. A. M. (2016). Um método para planejamento da produção em sistemas de manufatura flexível. In *Anais do XLVIII Simpósio Brasileiro de Pesquisa Operacional*, p. 3717–3724.
- Privault, C. e Finke, G. (2000). k-server problems with bulk requests: an application to tool switching in manufacturing. *Annals of Operations Research*, 96(1-4):255–269.
- Senne, E. L. F. e Yanasse, H. H. (2009). 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, p. 68–72.
- Shirazi, R. e Frizelle, G. (2001). Minimizing the number of tool switches on a flexible machine: an empirical study. *International Journal of Production Research*, 39(15):3547–3560.
- Tang, C. S. e Denardo, E. V. (1988). Models arising from a flexible manufacturing machine, part i: minimization of the number of tool switches. *Operations research*, 36(5):767–777.
- Yanasse, H. e Lamosa, M. (2005). An application of the generalized travelling salesman problem: the minimization of tool switches problem. In *International Annual Scientific Conference of the German Operations Research Society*, p. 90.
- Yanasse, H. e Lamosa, M. (2006). On solving the minimization of tool switches problem using graphs. In *Annals of the XII ICIEOM – International Conference on Industrial Engineering and Operations Management*.
- Yanasse, H. H. e Pinto, M. J. (2002). The minimization of tool switches problem as a network flow problem with side constraints. *XXXIV SBPO–Simpósio Brasileiro de Pesquisa Operacional, realizado no Rio de Janeiro, RJ*, 8:86.
- Yanasse, H. H., Rodrigues, R. d. C. M., e Senne, E. L. F. (2009). Um algoritmo enumerativo baseado em ordenamento parcial para resolução do problema de minimização de trocas de ferramentas. *Gestão and Produção*, 13(3).
- Zhou, B.-H., Xi, L.-F., e Cao, Y.-S. (2005). 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.