

Sistema de meta-aprendizado para a seleção de meta-heurísticas para o problema do caixeiro viajante

Jorge Yoshio Kanda¹

¹Universidade Federal do Amazonas
Instituto de Ciências Exatas e Tecnologia
Rua Nossa Senhora do Rosário, 3863 Tiradentes
69103-128 – Itacoatiara – Amazonas – Brasil

Resumo. *O Problema do Caixeiro Viajante (PCV) é um problema de otimização com muitas variações para as quais existem muitas meta-heurísticas (MH) capazes de gerar boas soluções. É difícil saber previamente a MH que produzirá a melhor solução para um dado PCV. Este artigo propõe uma abordagem baseada em meta-aprendizado para selecionar a MH mais promissora para novos exemplos de PCV. Essa abordagem usa técnicas de aprendizado de máquina para induzir meta-modelos a partir de um conjunto de meta-dados. Cada instância do conjunto de meta-dados é um exemplo de PCV descrito por características (meta-atributos) do problema e por um ranking de desempenho das MHs (meta-atributo alvo). Os meta-modelos induzidos são usados para indicar uma lista ordenada de MHs para novas instâncias. Os resultados experimentais mostram que a abordagem proposta é bastante promissora.*

Palavras-chave: *aprendizado de máquina; meta-atributos; ranqueamento de rótulos; problema de seleção de algoritmos.*

1. Introdução

O problema do caixeiro viajante (PCV) é descrito por um conjunto de cidades, as conexões entre pares de cidades e o custo de viagem entre as cidades conectadas. A melhor solução para um PCV é a rota de menor custo, que inicia em uma das cidades, visita as demais uma única vez e termina na cidade inicial [Gutin and Punnen 2002].

Para muitos exemplos de PCV a busca exaustiva pela melhor solução é bastante demorada. Um problema com 20 cidades completamente interconectadas possui em torno de $1,22 \times 10^{17}$ soluções factíveis. Uma busca exaustiva processada por um computador com capacidade de processar um bilhão de soluções por segundo encontraria a solução ótima após quase quatro anos. Certamente, um decisor não irá esperar esse tempo para definir a melhor rota para os veículos de sua empresa.

Para esse tipo de problema, uma boa solução, próxima à solução ótima, pode ser obtida por meta-heurísticas (MHs), que são métodos de otimização baseados em estratégias robustas para escapar das soluções ótimas locais [Gendreau and Potvin 2010]. Diferentes MHs geram as melhores soluções para diferentes instâncias do PCV, então como selecionar a mais promissora para um dado PCV? Esse questionamento é um típico problema de seleção de algoritmos [Rice 1976]. De acordo com o Teorema “*No Free Lunch*” [Wolpert and Macready 1997] selecionar sempre um mesmo algoritmo não produz bons resultados. Por outro lado, processar todos os algoritmos disponíveis para a instância desejada e depois selecionar aquele com a melhor solução seria muito custoso.

Dado esse contexto, as principais contribuições deste artigo são: i) descrição de uma abordagem baseada em meta-aprendizado para a seleção das MHs mais promissoras dadas novas instâncias do PCV; ii) indução dos modelos de meta-aprendizado usando diferentes algoritmos de aprendizado de máquina; iii) proposta de três conjuntos de meta-atributos preditivos para o PCV; iv) análise do tempo de geração dos meta-atributos.

O restante deste artigo está organizado conforme segue: a seção 2 apresenta a definição formal do PCV. A relação entre meta-aprendizado e o problema de seleção de algoritmos é brevemente descrita na seção 3. A proposta de três conjuntos de meta-atributos para o PCV é detalhada na seção 4. Resultados experimentais são relatados e analisados na seção 5. Finalmente, a seção 6 discorre sobre as considerações finais.

2. O Problema do Caixeiro Viajante - PCV

Formalmente, um PCV é definido por meio de um grafo $G = (V, E)$, em que $V = \{v_1, \dots, v_n\}$ é o conjunto de vértices e $E = \{\langle v_i, v_j \rangle: v_i, v_j \in V\}$ é o conjunto de arestas. Cada vértice $v_i \in V$ representa uma cidade e cada aresta $\langle v_i, v_j \rangle \in E$ conecta as cidades v_i e v_j . Um PCV é fortemente conectado se $\forall (v_i, v_j) \exists \langle v_i, v_j \rangle \in E$, caso contrário é fracamente conectado. Cada aresta $\langle v_i, v_j \rangle$ é associada a um valor c_{ij} , que é o custo de viagem entre v_i e v_j . Se $c_{ij} = c_{ji} \forall \langle v_i, v_j \rangle \in E$, o PCV é simétrico, caso contrário é assimétrico.

Existem várias aplicações no mundo real que são modeladas por meio do PCV, tais como: rota de transporte escolar, sequenciamento de DNA, manufatura de circuitos de microchips e inspeções em plataformas petrolíferas (Applegate et al., 2007).

As principais MHs usadas para o PCV são: Busca Tabu (BT) [Glover et al. 1993], *Greedy Randomized Adaptive Search Procedure* (GRASP) [Feo and Resende 1995], *Simulated Annealing* (SA) [Kirkpatrick et al. 1983], Algoritmos Genéticos (AGs) [Holland 1973] e, Otimização baseada em Colônia de Formigas (OCF) [Dorigo and Gambardella 1997].

3. Meta-aprendizado para a seleção de algoritmos

Meta-aprendizado surgiu a partir da seleção do algoritmo de AM mais promissor para um novo conjunto de dados. Um dos objetivos do meta-aprendizado é aprender regras que relacionam características das instâncias de um problema com o desempenho de um conjunto de algoritmos para essas instâncias [Brazdil et al. 2009].

Pesquisas relacionando meta-aprendizado com problemas de otimização surgiram há poucos anos. Para citar alguns, o modelo de AM que estima o desvio percentual da solução de cada algoritmo para o problema de alocação a partir de soluções obtidas com inicializações aleatórias [Smith-Miles 2008]. Abordagem SATzilla [Xu et al. 2008] que prevê o tempo de execução de algoritmos para resolver problemas de satisfatibilidade proposicional usando propriedades de exemplos similares.

Alguns estudos aplicam meta-aprendizado em PCV, tais como: uso de meta-atributos sobre a localização bidimensional das cidades em modelos de redes neurais para estimar o esforço dos algoritmos para obter a melhor solução [Smith-Miles et al. 2010]; modelo de classificação multirrótulo para instâncias de PCV rotuladas com as MHs que geram a melhor solução [Kanda et al. 2011a]; aplicação desse modelo usando diferentes abordagens de redes neurais [Kanda et al. 2011b]; e recomendação do *ranking* de MHs para exemplos de PCV em dois diferentes cenários de aplicação [Kanda et al. 2012].

A recomendação de *ranking* de MHs pode ser modelada como um problema de ordenação de rótulos [Fürnkranz et al. 2008]. Nesse caso, a tarefa de aprendizado consiste em mapear as instâncias x de um conjunto de dados X para *rankings* \succ_x sobre um conjunto finito de rótulos $\mathcal{L} = \{\lambda_1, \dots, \lambda_m\}$, em que $\lambda_i \succ_x \lambda_j$ significa que para a instância x , o rótulo λ_i é preferível ao rótulo λ_j . Um *ranking* pode ser visto como uma permutação de \mathcal{L} e há apenas uma permutação τ tal que $\lambda_i \succ_x \lambda_j$ se, e somente se, $\tau(\lambda_i) < \tau(\lambda_j)$, em que $\tau(\lambda_i)$ denota a posição do rótulo λ_i no *ranking*. Mais detalhes sobre ordenação de rótulos podem ser encontrados em [Vembu and Gärtner 2011].

A abordagem de meta-aprendizado descrita neste artigo inicia com a obtenção de muitas instâncias do PCV a serem resolvidas por diferentes MHs. A ordem de desempenho das MHs para cada instância é o meta-atributo alvo. Características (meta-atributos preditivos) do problema e o meta-atributo alvo formam o conjunto de meta-dados. Uma técnica de AM deve ser escolhida para induzir o modelo preditivo (meta-modelo) a partir do conjunto de meta-dados. Como cada técnica tem seu viés indutivo [Mitchell 1997], três delas foram investigadas nos experimentos: rede *multilayer perceptron* (MLP), *K-Nearest-Neighbor* (K-NN) e árvore de decisão (AD) [Tan et al. 2006]. A capacidade preditiva de um meta-modelo pode ser medida por uma taxa de acurácia de *ranking*, como a Correlação de Spearman [Spearman 1904], a fim de comparar o *ranking* predito com o *ranking* conhecido. Segundo [Brazdil et al. 2009], esse desempenho depende dos meta-atributos. Os meta-atributos investigados neste estudo são detalhados na próxima seção.

4. Meta-atributos para o PCV

Os meta-atributos gerados a partir das características do PCV são descritos a seguir:

4.1. Medidas baseadas em custo de aresta e de vértice

Como um PCV pode ser representado por um grafo, medidas de arestas e vértices podem ser bons meta-atributos. Os meta-atributos (*MVA*) usados nos experimentos estão listados na Tabela 1.

O custo de uma aresta (C_i^E) é extraído do grafo e o custo de um vértice (C_j^V) é a média dos custos das arestas conectadas ao vértice. Em geral, MHs fornecem uma boa solução após comparar diferentes soluções contidas na região de vizinhança da solução corrente. No PCV, o número de vizinhos depende algumas vezes do número de cidades do problema. Por isso, medidas relacionadas aos custos dos vértices foram calculadas. Além disso, uma solução composta por n cidades é geralmente constituída por muitas arestas de baixo custo, indicando a relevância de extrair propriedades dos custos das arestas.

4.2. Medidas baseadas em características das MHs

Antes de detalhar o primeiro conjunto de meta-atributos, alguns conjuntos de soluções para uma instância i do PCV com n cidades devem ser definidos.

- $S_i = \{s_i^1, \dots, s_i^z\}$: conjunto de todas as soluções factíveis;
- $R_i = \{r_i^1, \dots, r_i^n\}$: conjunto de n soluções aleatórias, tal que $R_i \subset S_i$;
- $G_i = \{g_i^1, \dots, g_i^n\}$: conjunto de n soluções geradas pela heurística gulosa a partir de diferentes cidades iniciais, tal que $G_i \subset S_i$;
- $N_i^j = \{n_i^{j1}, \dots, n_i^{jn}\}$: conjunto de soluções vizinhas a s_i^j .

Tabela 1. Meta-atributos (MA) baseados em custos de aresta e vértice.

MA	Notação Matemática	Descrição
V_{number}	$length(V)$	número de vértices (n)
C_{min}^V	$min\{C_1^V, \dots, C_n^V\}$	menor custo de vértice
C_{max}^V	$max\{C_1^V, \dots, C_n^V\}$	maior custo de vértice
C_{avg}^V	$\sum_{i=1}^n C_i^V / n$	média dos custos dos vértices
C_{sd}^V	$\sqrt{\frac{1}{n-1} \sum_{i=1}^n (C_i^V - C_{avg}^V)^2}$	desvio padrão dos custos dos vértices
C_{median}^V	$median\{C_1^V, \dots, C_n^V\}$	mediana dos custos dos vértices
C_{nn}^V	$\sum_{i=1}^n min\{c_{i1}, \dots, c_{in}\}$	custo total dos vértices para o mais próximo
E_{number}	$length(E)$	número de arestas (m)
C_{min}^E	$min\{C_1^E, \dots, C_m^E\}$	menor custo de aresta
C_{max}^E	$max\{C_1^E, \dots, C_m^E\}$	maior custo de aresta
C_{avg}^E	$\sum_{i=1}^m C_i^E / m$	média dos custos das arestas
C_{sd}^E	$\sqrt{\frac{1}{m-1} \sum_{i=1}^m (C_i^E - C_{avg}^E)^2}$	desvio padrão dos custos das arestas
C_{median}^E	$median\{C_1^E, \dots, C_m^E\}$	mediana dos custos das arestas
C_{lowest}^E	$\sum_{i=1}^n ord(C_i^E)$	soma dos n menores custos de arestas

Neste trabalho, uma solução vizinha n_i^{jk} é definida como uma permutação entre duas cidades adjacentes pertencentes à solução s_i^j . O valor da solução s_i^j é dado por $c(s_i^j)$ e o de sua melhor solução vizinha, n_i^{j*} , é dado por $c(n_i^{j*}) = \min\{c(n_i^{j1}), \dots, c(n_i^{jn})\}$.

A Tabela 2 descreve o conjunto de meta-atributos (*PMH*) gerados a partir das propriedades das MHs. Durante a execução de BT e SA, uma solução vizinha pior pode ser aceita para escapar das soluções ótimas locais. Os três primeiros meta-atributos da Tabela 2 comparam a qualidade de cada solução aleatória com a melhor solução contida na região da vizinhança. Meta-atributo *IMV* mensura a taxa da melhor solução vizinha ser melhor que a solução aleatória. A função $menor(c(x), c(y)) = 1$ se $c(x) < c(y)$, e 0 caso contrário. Meta-atributo *QMV* fornece a média entre o custo da melhor solução vizinha e o custo da rota aleatória. Meta-atributo *QSV* indica a taxa das soluções vizinhas serem melhores que a solução aleatória.

Uma solução fornecida pelo GRASP é construída a cada iteração com a inserção de uma nova cidade à rota. Essa inserção depende dos parâmetros de gula e aleatoriedade. Os meta-atributos *IGS* e *RGA* comparam soluções gulosas com soluções aleatórias. O primeiro calcula o índice de soluções gulosas serem melhores que as soluções aleatórias, e o outro soma as razões entre o custo da solução gulosa e o custo da solução aleatória.

Os próximos três meta-atributos capturam propriedades dos algoritmos genéticos. Usando as soluções aleatórias, p pares dessas soluções são escolhidos ao acaso para compor o conjunto de soluções-pai, $\hat{R}_i = \{(\hat{r}_{i1}^1, \hat{r}_{i2}^1), \dots, (\hat{r}_{i1}^p, \hat{r}_{i2}^p)\}$. Duas soluções-pai são combinadas para gerar as soluções-filho, $\check{R}_i = \{(\check{r}_{i1}^1, \check{r}_{i2}^1), \dots, (\check{r}_{i1}^p, \check{r}_{i2}^p)\}$. O meta-atributo *QMF* mensura a taxa da melhor solução-filho ter um custo de rota menor que a melhor solução-pai. A razão média entre os custos das soluções-filho e os custos das soluções-pai é mensurada pelo meta-atributo *RPF*. A taxa das soluções reproduzidas serem melhores do que aquelas usadas para reproduzi-las é fornecida pelo meta-atributo *QRS*.

Tabela 2. Meta-atributos (MA) baseados em medidas de características das MHs.

MA	Notação Matemática	Descrição
<i>IMV</i>	$\frac{1}{n} \sum_{k=1}^n \text{menor}(c(n_i^{j*}), c(r_i^k))$	índice da melhor vizinha
<i>QMV</i>	$\frac{1}{n} \sum_{k=1}^n \frac{c(n_i^{j*})}{c(r_i^k)}$	relação entre a solução gerada e a melhor vizinha
<i>QSV</i>	$\frac{1}{n^2} \sum_{j=1}^n \sum_{k=1}^n \text{menor}(c(n_i^{j*}), c(r_i^j))$	relação entre a solução gerada e todas as soluções vizinhas
<i>IGS</i>	$\frac{1}{n^2} \sum_{j=1}^n \sum_{k=1}^n \text{menor}(c(g_i^j), c(r_i^k))$	índice da solução gulosa
<i>RGA</i>	$\sum_{j=1}^n \frac{c(g_i^j)}{c(r_i^j)}$	relação entre solução gulosa e solução aleatória
<i>QMF</i>	$\frac{1}{p} \sum_{j=1}^p \text{menor}(c((\check{R}_i^j)^*), c(\hat{R}_i^j)^*)$	média da melhor solução-filho superar a melhor solução-pai
<i>RPF</i>	$\frac{1}{p} \sum_{j=1}^p \frac{c((\hat{R}_i^j)^*)}{c((\check{R}_i^j)^*)}$	relação entre a melhor solução-pai e a melhor solução-filho
<i>QRS</i>	$\frac{1}{4p} \sum_{j=1}^p \sum_{k=1}^2 \sum_{l=1}^2 \text{menor}(c(\check{r}_{ik}^j), c(\hat{r}_{il}^j))$	qualidade das soluções-filho
<i>IAC</i>	$\frac{1}{n} \sum_{j=1}^n \sum_{k=j+1}^n \text{tamanho}(E_i^j \cap E_i^k)$	taxa de arestas compartilhadas
<i>AMC</i>	$\frac{1}{n} \sum_{j=1}^n \text{pertence}(e_{mc}, g_s_i^j)$	taxa de soluções que contém a aresta mais comum

Uma solução fornecida pela OCF indica a rota final das formigas, que iniciam suas rotas em cidades diferentes. No final, as rotas compartilham as mesmas arestas porque as formigas mudam seu trajeto ao longo do tempo, seguindo pelo caminho onde passam mais formigas. Seja $E_i^j = \{e_{i1}^j, \dots, e_{in}^j\}$ o conjunto de arestas da solução s_i^j , o meta-atributo *IAC* mostra o índice de arestas compartilhadas entre todos os pares de soluções gulosas. A função $\text{tamanho}(E)$ retorna o número de elementos do conjunto E . Uma função de mapeamento $q : E \rightarrow \mathfrak{R}$ identifica o número de soluções gulosas que contém cada aresta, assim a aresta mais compartilhada (e_{mc}) é aquela em que $q(e_{mc}) = \max \{q(e_1), \dots, q(e_m)\}$. Finalmente, o meta-atributo *AMC* mensura a frequência de participação da aresta e_{mc} nas soluções gulosas. A função $\text{pertence}(a, X) = 1$ se $a \in X$, e 0 caso contrário.

4.3. Medidas baseadas em *landmarkers*

Os meta-atributos baseados em *landmarkers* (*CLM*) estimam o desempenho das MHs ao executar algoritmos das mesmas em uma versão simplificada [Brazdil et al. 2009].

Sem a intenção de promover qualquer MH, mas garantir que soluções sejam obtidas rapidamente, seus parâmetros foram configurados conforme segue: BT: tamanho da lista tabu = 2, número de vizinhos = 2 e iterações sem melhoria da solução = 1; GRASP: número de iterações = 1 e nível de aleatoriedade e gula = 0,5; SA: temperatura inicial = 1, taxa de aumento da temperatura = 10%, taxa de aceitação da solução vizinha = 90% e taxa de resfriamento = 1%; AG: tamanho da população = 4, pais selecionados por torneio, taxa de cruzamento = 100%, operador de cruzamento = PMX (*Partial Mapped Crossover*) [Goldberg and Lingle 1985], taxa de mutação = 5% e seleção da próxima população por elitismo; e OCF: número de formigas = 2, taxa de evaporação do feromônio = 50%, taxa de influência do feromônio e da informação da heurística = 1.

Como essas MHs são estocásticas cada instância x do PCV foi submetida 30 vezes em cada algoritmo com diferentes inicializações e a média das 30 soluções foi atribuída como desempenho da MH para essa instância. A Tabela 3 mostra os meta-atributos gerados a partir dos *landmarkers*.

Tabela 3. Meta-atributos extraídos a partir da caracterização de *landmarkers*.

Meta-atributos	Notação Matemática	Descrição
<i>DBT</i>	$tabu(x)$	desempenho BT
<i>DGR</i>	$grasp(x)$	desempenho GRASP
<i>DSA</i>	$sa(x)$	desempenho SA
<i>DAG</i>	$genetico(x)$	desempenho AG
<i>DCF</i>	$formiga(x)$	desempenho OCF

5. Experimentos

Além da descrição dos experimentos realizados para avaliar a abordagem de meta-aprendizado para a seleção de MHs, esta seção analisa os resultados obtidos.

5.1. Configuração experimental

Foram avaliados 4 cenários de PCV: simétrico fortemente conectado, assimétrico fortemente conectado, simétrico fracamente conectado e assimétrico fracamente conectado.

Para cada cenário foram gerados 600 subproblemas a partir de arquivos contidos na biblioteca TSPLIB [Reinelt 1991]. De cada um dos arquivos: *eil76*, *ch150*, *lin318* e *u724*, foram gerados 150 subproblemas simétricos com 10, 25, 50 e 100 cidades, respectivamente. Os subproblemas assimétricos com essas quantidades de cidades foram gerados a partir dos arquivos: *p43*, *kro124p*, *ftv170* e *rbg443*, respectivamente. Os custos originais entre as cidades foram mantidos nos subproblemas.

Os subproblemas fracamente conectados foram gerados usando os fortemente conectados, nos quais foram aplicados o modelo mundo pequeno [Watts 1999]. Esse modelo gera uma rede em que a maioria dos vértices pode ser alcançada partindo dos demais vértices usando poucas arestas. Neste estudo, cada vértice foi conectado aos seus k -vizinhos mais próximos, em que k é igual a 60% do número de cidades do problema. Em seguida, cada aresta foi aleatoriamente reconectada com uma probabilidade $p = 0,1$. Os valores para k e p foram escolhidos seguindo a abordagem descrita em [Costa et al. 2007].

Todos os subproblemas gerados foram submetidos às MHs selecionadas: BT, GRASP, SA, AG e OCF. Os parâmetros a seguir foram definidos após as MHs encontrarem soluções razoáveis em experimentos preliminares. BT: tamanho da lista tabu = 2 e iterações sem melhoria na solução = 2; GRASP: número de iterações = 10 e nível de gula e aleatoriedade = 0.5; SA: temperatura inicial = 0.1, taxa de aumento da temperatura = 10%, taxa de aceitação da solução vizinha = 90% e taxa de resfriamento = 1%; AG: tamanho da população = 20, pais selecionados por torneio, taxa de cruzamento = 100%, operador de cruzamento = PMX, taxa de mutação = 5% e seleção da nova população por elitismo. OCF: número de formigas = 5, taxa de evaporação do feromônio = 50%, taxa de influência do feromônio = 1 e taxa da informação da heurística = 1.

Para cada subproblema, cada MH foi executada 30 vezes (com semente inicial diferente), produzindo 30 soluções. O mesmo tempo de processamento foi adotado para todas as execuções como critério de parada dos algoritmos. A média das 30 soluções foi utilizada como desempenho da MH e o *ranking* de desempenho das MHs foi atribuído ao meta-atributo alvo para o meta-exemplo que corresponde a um subproblema gerado.

Os meta-modelos baseados nas técnicas MLP, AD e K-NN foram implementados, respectivamente, a partir dos pacotes *nnet*, *FNN* e *Clus* na Linguagem de Programação R. Não houve alteração dos valores sugeridos para os parâmetros livres. Na fase de treinamento dos meta-modelos, foi adotada a metodologia de validação cruzada em 10 pastas [Tan et al. 2006]. As redes MLP foram treinadas com uma camada escondida cujo número de neurônios foi definido conforme a proposta descrita em [Haykin 2009].

5.2. Resultados experimentais

A capacidade preditiva dos meta-modelos foi mensurada por meio da comparação entre o *ranking* de MHs predito e o *ranking* previamente conhecido para cada instância de teste. O Coeficiente de Spearman (r_S) foi calculado para cada par de *ranking* $\langle \text{predito}, \text{ideal} \rangle$ e a média (\bar{r}_S) foi adotada como desempenho preditivo. O desempenho de cada meta-modelo foi comparado com o desempenho de um modelo baseline, que recomenda o *ranking* médio para todas as instâncias de teste.

A diferença significativa entre o desempenho do meta-modelo e o do baseline foi avaliada estatisticamente por meio do teste t, seguindo a recomendação descrita em [Demšar 2006]. Para esse teste, o valor p indica a probabilidade de obter resultados tão extremos quanto àqueles observados, assumindo como verdade a hipótese nula (resultados \bar{r}_S iguais). Geralmente, rejeita-se a hipótese nula se $p \leq \alpha$ (nível de significância adotado). Neste estudo, foi usado $\alpha = 5\%$ para todas as avaliações. Esse teste com o mesmo nível de significância foi adotado para identificar possível diferença entre o desempenho preditivo dos meta-modelos quando usados diferentes conjuntos de meta-atributos.

5.2.1. Cenário simétrico fortemente conectado

A Tabela 4 mostra a acurácia de *ranking* (\bar{r}_S) dos meta-modelos para cada conjunto de meta-atributos proposto. Os resultados do teste t mostram que todos os meta-modelos tiveram um desempenho preditivo melhor que o modelo baseline ($\bar{r}_S = 0.89$).

Tabela 4. Acurácia de *ranking* no cenário simétrico fortemente conectado.

Meta-atributos	MLP	K-NN	AD
<i>MVA</i>	0,96 ± 0,12	0,95 ± 0,09	0,95 ± 0,09
<i>PMH</i>	0,94 ± 0,11	0,95 ± 0,07	0,96 ± 0,09
<i>CLM</i>	0,95 ± 0,09	0,95 ± 0,10	0,96 ± 0,07

Os desempenhos preditivos dos meta-modelos foram similares independente do conjunto de meta-atributos usado. O teste t detectou apenas diferença significativa no desempenho dos meta-modelos MLP ao comparar os seguintes pares dos conjuntos de meta-atributos: (*MVA*, *PMH*) e (*MVA*, *CLM*).

5.2.2. Cenário assimétrico fortemente conectado

Em relação ao cenário anterior, o número de diferentes *rankings* de MHs foi maior e a frequência do *ranking* majoritário foi menor. Essas situações reduziram o desempenho preditivo (0,59) do modelo baseline. Por outro lado, os meta-modelos continuaram com uma boa capacidade preditiva conforme mostra a Tabela 5. O teste estatístico adotado apontou uma superioridade preditiva de todos os meta-modelos em relação ao baseline.

Tabela 5. Acurácia de *rankng* no cenário assimétrico fortemente conectado.

Meta-atributos	MLP	K-NN	AD
<i>MVA</i>	0,94 ± 0,12	0,93 ± 0,12	0,94 ± 0,12
<i>PMH</i>	0,91 ± 0,17	0,93 ± 0,13	0,93 ± 0,13
<i>CLM</i>	0,93 ± 0,14	0,93 ± 0,15	0,93 ± 0,14

Os resultados experimentais também indicaram que somente o desempenho dos meta-modelos induzidos a partir MLP foi significativamente diferente quando foram usados os conjuntos de meta-atributos *MVA* e *PMH*.

5.2.3. Cenário simétrico fracamente conectado

As MHs nem sempre conseguem encontrar uma solução factível para instâncias de PCV fracamente conectadas. Nos experimentos realizados, o viés da meta-heurística OCF foi primordial para obter uma solução factível para a maioria dos subproblemas desse tipo. Esta razão propiciou o desempenho do modelo baseline ter um valor igual a 0,81. Os valores do desempenho preditivo dos meta-modelos estão mostrados na Tabela 6.

Tabela 6. Acurácia de *rankng* no cenário simétrico fracamente conectado.

Meta-atributos	MLP	K-NN	AD
<i>MVA</i>	0,93 ± 0,14	0,93 ± 0,14	0,92 ± 0,15
<i>PMH</i>	0,92 ± 0,16	0,93 ± 0,14	0,92 ± 0,15
<i>CLM</i>	0,92 ± 0,18	0,92 ± 0,16	0,93 ± 0,15

Resultado do teste t comprovou o melhor desempenho dos meta-modelos em relação ao baseline. Não foi detectado diferença significativa no desempenho dos meta-modelos induzidos pela mesma técnica de AM em diferentes conjuntos de meta-atributos.

5.2.4. Cenário assimétrico fracamente conectado

Para este cenário, os meta-modelos tiveram desempenho menor que nos cenários anteriores (Tabela 7). A média da acurácia de *ranking* dos meta-modelos foi menor que 0,90 e o desempenho do baseline igual a 0,77. O teste t indicou que o desempenho do meta-modelo foi melhor que o modelo baseline em três situações: MLP com meta-atributos *PMH*, MLP com meta-atributos *CLM* e K-NN com meta-atributos *CLM*.

Tabela 7. Acurácia de *rankng* no cenário assimétrico fracamente conectado.

Meta-atributos	MLP	K-NN	AD
<i>MVA</i>	0,83 ± 0,27	0,85 ± 0,25	0,86 ± 0,25
<i>PMH</i>	0,80 ± 0,33	0,83 ± 0,28	0,86 ± 0,23
<i>CLM</i>	0,78 ± 0,27	0,79 ± 0,27	0,80 ± 0,24

O aumento no valor do desvio padrão indica que os meta-modelos tiveram desempenho fraco na predição de alguns meta-exemplos. Esse resultado decorre da maior variação na qualidade das soluções geradas pelas MHs para subproblemas fracamente conectados, principalmente aquelas com menos cidades. Nesses casos, é mais difícil encontrar uma solução factível quando não existem algumas arestas, prejudicando o desempenho das MHs, principalmente aquelas que iniciam com uma solução aleatória.

Diferenças significativas no desempenho dos meta-modelos foram detectadas entre o conjunto de meta-atributos *CLM* e cada um dos outros conjuntos, independente da técnica de AM aplicada. Além disso, foi também detectada diferença significativa no desempenho da MLP quando usados os conjuntos de meta-atributos *MVA* e *PMH*.

Os meta-atributos *CLM* não foram tão adequados para induzir os meta-modelos, pois as MHs não foram capazes de obter uma solução rápida para vários subproblemas. Em relação aos meta-atributos *PMH*, alguns dependem de soluções aleatórias e de suas respectivas soluções vizinhas. Nem sempre é possível obter aleatoriamente uma solução factível para instâncias fracamente conectadas. Por isso, os meta-modelos induzidos a partir de MLP tiveram dificuldades para prever o valor do meta-atributo de saída.

5.3. Análise do tempo de geração dos meta-atributos

Nos cenários investigados, os meta-modelos tiveram bom desempenho preditivo usando os conjuntos de meta-atributos descritos. No entanto, é importante estimar o tempo computacional para gerar os meta-atributos. Se o tempo para calcular todos os meta-atributos de um conjunto para uma dada instância de PCV for maior que o tempo para executar todas as MHs, então o uso de meta-modelos não propicia qualquer vantagem.

Para investigar esse aspecto, a soma do tempo que todas as MH forneceram sua solução foi comparada com o tempo para gerar os diferentes conjuntos de meta-atributos. O tempo de treinamento dos meta-modelos não foi considerado porque o mesmo ocorre apenas uma vez. Usando o meta-modelo treinado, o tempo de processamento dos dados fornecidos em sua entrada é bastante rápido.

A fim de ilustrar a notação adotada, considere que todas as MHs fornecem as soluções em 2 segundos e os meta-atributos são calculados em 0,4 segundos. Nesse caso, o tempo economizado por gerar os meta-atributos é de 80%. Por outro lado, se os meta-atributos são gerados em 5 segundos, o tempo não economizado é igual a -150%. O valor negativo indica perda de tempo ao usar a abordagem de meta-aprendizado.

Nos experimentos realizados, somente os conjuntos de meta-atributos *MVA* e *CLM* foram gerados mais rápido que a execução das MHs, principalmente, para as instâncias com mais cidades. A geração dos meta-atributos *MVA* foi a mais rápida por causa do uso de medidas que requerem cálculos simples.

A Tabela 8 mostra a taxa de tempo economizado ao gerar os meta-atributos nos dois cenários fortemente conectado. Os meta-atributos *MVA* foram gerados rapidamente, usando no máximo 20% do tempo para executar todas as MHs disponíveis. Para os meta-atributos *CLM*, não houve um ganho de tempo para subproblemas menores, pois tanto a versão original quanto a simplificada das MHs encontraram rapidamente boas soluções. Para as instâncias maiores, não houve uma significativa economia de tempo porque as MHs construtivas (ex.: GRASP) precisaram praticamente do mesmo tempo para construir a sua solução independente da versão aplicada. O tempo para gerar os meta-atributos dos conjuntos *PMH* e *CLM* aumentou com o número de cidades do problema por causa do custo computacional quadrático ao gerar alguns desses meta-atributos.

Tabela 8. Economia de tempo na geração de meta-atributos — cenário fortemente conectado.

a) simétrico				b) assimétrico			
Tamanho	<i>MVA</i>	<i>PMH</i>	<i>CLM</i>	Tamanho	<i>MVA</i>	<i>PMH</i>	<i>CLM</i>
10 cidades	93%	81%	0%	10 cidades	93%	85%	0%
25 cidades	93%	44%	12%	25 cidades	92%	47%	15%
50 cidades	92%	-31%	26%	50 cidades	92%	3%	26%
100 cidades	82%	-469%	19%	100 cidades	80%	-394%	22%

As taxas de tempo economizado por gerar os meta-atributos para os cenários fracamente conectado estão mostrados na Tabela 9. Independente do problema ser simétrico ou assimétrico, o tempo economizado na geração dos meta-atributos dependem apenas do número de cidades. Os meta-atributos *MVA* foram gerados mais rapidamente devido a ausência de arestas nos subproblemas para esses cenários. Por causa disso, houveram poucas operações nos cálculos dos meta-atributos baseados em custos de aresta e vértice.

Tabela 9. Economia de tempo na geração de meta-atributos — cenário fracamente conectado.

a) simétrico				b) assimétrico			
Tamanho	<i>MVA</i>	<i>PMH</i>	<i>CLM</i>	Tamanho	<i>MVA</i>	<i>PMH</i>	<i>CLM</i>
10 cidades	97%	90%	11%	10 cidades	97%	0%	12%
25 cidades	97%	66%	14%	25 cidades	96%	66%	14%
50 cidades	96%	20%	28%	50 cidades	97%	28%	23%
100 cidades	91%	-277%	20%	100 cidades	92%	-228%	18%

6. Considerações Finais

O problema de recomendar as MHs mais promissoras para uma dada instância de PCV foi encaminhado por uma abordagem de meta-aprendizado, que consistiu em induzir modelos de AM para aprender a relação entre propriedades do PCV e o desempenho das MHs. Os meta-modelos foram induzidos usando as técnicas MLP, K-NN e árvore de decisão para ordenação de rótulos. Três conjuntos de meta-atributos foram avaliados: medidas de vértices e arestas, propriedades de meta-heurísticas e caracterização de *landmarkers*.

Resultados experimentais em quatro cenários mostraram que os meta-modelos produziram bons *rankings* de MHs para novas instâncias do PCV. Seguindo o *ranking* recomendado, podem ser obtidas melhores soluções para o PCV do aquelas indicadas por uma estratégia simples como a recomendação baseada no *ranking* médio.

Para cada cenário, o tempo médio para gerar cada conjunto de meta-atributos foi comparado com o tempo de execução de todas as MHs candidatas. Como alguns meta-atributos tem complexidade computacional quadrática, o custo para processá-los foi maior do que executar as MHs, principalmente, para as instâncias maiores. Nesse aspecto, somente os conjuntos de meta-atributos baseados em medidas de vértices e de arestas e em propriedades de *landmarkers* tiveram resultados satisfatórios.

Em trabalhos futuros, o desempenho preditivo do meta-modelo pode ser avaliado em cenários diferentes do que tenha sido treinado. Além disso, esta abordagem de meta-aprendizado pode ser avaliada em outros problemas de otimização do mundo real.

Agradecimentos

FAPEAM, CAPES, Prof. André Carvalho, Prof. Eduardo Hruschka, Prof. Carlos Soares e os revisores que avaliaram este artigo.

Referências

- Brazdil, P., Giraud-Carrier, C., Soares, C., and Vilalta, R. (2009). *Metalearning: Application to Data Mining*. Springer, Berlin, Germany.
- Costa, L., Rodrigues, F., Travieso, G., and Boas, P. V. (2007). Characterization of complex networks: A survey of measurements. *Advances in Physics*, 56:167–242.
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30.
- Dorigo, M. and Gambardella, L. M. (1997). Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66.
- Feo, T. and Resende, M. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133.
- Fürnkranz, J., Hüllermeier, E., Mencía, E., and Brinker, K. (2008). Multilabel classification via calibrated label ranking. *Mach. Learn.*, 73:133–153.
- Gendreau, M. and Potvin, J.-Y. (2010). *Handbook of Metaheuristics*. Springer, New York, 2a. edition.
- Glover, F., Taillard, E., and Taillard, E. (1993). A user’s guide to tabu search. *Annals of Operations Research*, 41:1–28.
- Goldberg, D. E. and Lingle, R. (1985). Alleles, loci, and the traveling salesman problem. In Grefenstette, J. J., editor, *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*. Lawrence Erlbaum Associates, Publishers.
- Gutin, G. and Punnen, A. (2002). *The Traveling Salesman Problem and Its Variations*. Kluwer Academic Publishers, Netherlands.

- Haykin, S. S. (2009). *Neural networks and learning machines*. Pearson Education inc, New York, 3a. edition.
- Holland, J. (1973). Genetic algorithms and the optimal allocations of trial. *SIAM Journal of Computing*, 2:88–105.
- Kanda, J., De Carvalho, A., Hruschka, E., and Soares, C. (2011a). Selection of algorithms to solve traveling salesman problems using meta-learning. *Int. J. Hybrid Intell. Syst.*, 8(3):117–128.
- Kanda, J., Soares, C., Hruschka, E., and De Carvalho, A. (2012). A meta-learning approach to select meta-heuristics for the traveling salesman problem using MLP-based label ranking. In *Proceedings of the 2012 Nineteenth International Conference on Neural Information Processing, ICONIP '12*, pages 488–495, Berlin. Springer-Verlag.
- Kanda, J. Y., De Carvalho, A. C., Hruschka, E. R., and Soares, C. (2011b). Using meta-learning to recommend meta-heuristics for the traveling salesman problem. *Machine Learning and Applications, Fourth International Conference on*, 1:346–351.
- Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). Optimization by simulated annealing. *Science*, 220:671–680.
- Mitchell, T. (1997). *Machine Learning*. McGraw Hill, New York.
- Reinelt, G. (1991). TSPLIB - a traveling salesman problem library. *Inform Journal on Computing*, 3(4):376–384.
- Rice, J. (1976). The algorithm selection problem. *Advances in Computers*, 15:65–118.
- Smith-Miles, K., van Hemert, J., and Lim, X. Y. (2010). Understanding tsp difficulty by learning from evolved instances. In *Proceedings of the 4th international conference on Learning and intelligent optimization, LION'10*, pages 266–280, Berlin, Heidelberg. Springer-Verlag.
- Smith-Miles, K. A. (2008). Towards insightful algorithm selection for optimisation using meta-learning concepts. In *Proceedings of the IEEE International Joint Conference on Neural Networks 2008*, pages 4118–4124.
- Spearman, C. (1904). The proof and measurement of association between two things. *American Journal of Psychology*, 15:72–101.
- Tan, P.-N., Steinbach, M., and Kumar, V. (2006). *Introduction to Data Mining*. Addison Wesley, Boston.
- Vembu, S. and Gärtner, T. (2011). Label ranking algorithms: A survey. In Fürnkranz, J. and Hüllermeier, E., editors, *Preference Learning*, pages 45–64. Springer, Berlin Heidelberg.
- Watts, D. (1999). *Small worlds: the dynamics of networks between order and randomness*. Princeton University Press,, Princeton.
- Wolpert, D. and Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1:67–82.
- Xu, L., Hutter, F., Hoos, H. H., and Leyton-Brown, K. (2008). SATzilla: portfolio-based algorithm selection for SAT. *Journal of Artificial Intelligence Research*, 32(1):565–606.