

IFNMG - Instituto Federal do Norte de Minas Gerais
Curso: Ciência da Computação
Disciplina: Técnicas de Busca Heurística

Trabalho 002

Artur Pereira Neto
Breno Vambáster Cardoso Lima

Introdução

Este trabalho apresenta os resultados da investigação com busca local e tabu em um problema de minimização de distâncias de pontos e seus respectivos centroides.

Contextualização do problema

Este trabalho utiliza a base de dados Wine disponibilizada pelo UCI Machine Learning Repository. Essa base de dados é amplamente empregada em tarefas de aprendizado de máquina, incluindo classificação, agrupamento e análise de características químicas. Ela contém informações sobre os componentes químicos de vinhos oriundos de três variedades distintas cultivadas na região italiana de Lombardia. O objetivo principal dessa base é possibilitar a classificação dos vinhos em uma das três categorias, com base nas características químicas fornecidas. A base de dados foi originalmente publicada por Forina et al., no estudo intitulado "Chemical and Sensory Analysis of Italian Wines", que explorou como os componentes químicos do vinho poderiam ser utilizados para diferenciá-los de acordo com suas variedades. Essa base é uma referência na literatura devido à sua qualidade e relevância para estudos em classificação e agrupamento de dados multidimensionais. A base de dados contém 3 classes distintas, cada uma correspondendo a uma variedade de vinho. Cada registro é descrito por 13 atributos (contínuos e discretos) que descrevem as características físico-químicas do vinho. Nesse estudo a fim de reduzir a complexidade do problema a dimensionalidade da base de dados foi reduzida para um domínio em R^2 . Os parâmetros Flavonoides e Fenóis totais foram as características escolhidas para a simplificação da base. Semelhante à primeira etapa da investigação

Objetivos

Aplicar técnicas de busca local para otimização do resultado da primeira etapa da investigação. Previamente, o algoritmo de K-means havia sido aplicado na base de dados para encontrar um conjunto de 3 centroides que minimizasse o somatório das distâncias euclidianas dos registros às suas respectivas coordenadas.

Conceitos principais

Busca Local

A técnica heurística de busca local é um método utilizado em problemas de otimização, no qual se busca iterativamente uma solução melhor a partir de uma solução inicial, explorando soluções vizinhas e fazendo ajustes incrementais. O objetivo é melhorar uma solução existente, movendo-se para soluções "vizinhas" que apresentam uma qualidade superior, com a esperança de encontrar a solução ótima ou uma solução suficientemente boa para o problema.

No contexto da busca local, existem duas abordagens principais que se concentram em buscar uma melhoria: a primeira melhora e a melhor melhora.

1. Busca Local com Primeira Melhora:

Nesse tipo de busca, a cada iteração, o algoritmo explora as soluções vizinhas da solução atual. Se encontrar uma solução vizinha que seja melhor que a solução atual, ele faz a troca imediatamente e passa para a nova solução. Não se continua a busca para outras soluções vizinhas após encontrar a primeira melhoria, ou seja, a busca local com primeira melhora é mais "agressiva" e toma decisões rápidas. Isso pode ser vantajoso quando se busca uma solução em um curto espaço de tempo, mas pode ser menos eficaz em encontrar soluções ótimas, pois pode levar a soluções subótimas e ficar preso em ótimos locais (ou "mínimos locais").

2. Busca Local com Melhor Melhora:

Na busca local com melhor melhora, o algoritmo também explora as soluções vizinhas, mas, ao contrário da primeira melhora, ele examina todas as soluções vizinhas antes de tomar uma decisão. Em seguida, escolhe a melhor solução entre as vizinhas avaliadas e a adota como a nova solução atual. Isso aumenta as chances de encontrar uma solução mais próxima do ótimo, já que o algoritmo realiza uma busca mais completa antes de fazer a troca, mas, por outro lado, pode exigir mais tempo computacional, dado que todas as soluções vizinhas precisam ser analisadas.

Ambas as abordagens são tipos de busca local porque, em ambos os casos, o algoritmo busca iterativamente melhorar a solução com base apenas nas soluções vizinhas da solução atual. No entanto, a diferença está na estratégia de escolha da solução vizinha a ser adotada: a busca com primeira melhora toma a decisão rapidamente, enquanto a busca com melhor melhora toma uma decisão mais ponderada, analisando todas as opções antes de escolher a melhor. Em muitos problemas de otimização, a busca local pode ser útil para encontrar boas soluções em um tempo razoável, especialmente quando o espaço de busca é grande e o algoritmo de busca exata seria computacionalmente caro. Porém, ambos os métodos podem ficar presos em ótimos locais, sendo necessárias técnicas adicionais, como busca em múltiplos inícios ou algoritmos de escape de ótimos locais, para melhorar o desempenho em encontrar soluções ótimas globais.

Busca tabu

A técnica heurística de busca tabu é uma abordagem de otimização que visa explorar eficientemente o espaço de soluções de um problema, evitando que o algoritmo fique preso em ótimos locais (ou mínimos locais). Essa técnica é particularmente útil em problemas complexos, onde o espaço de soluções é grande e difícil de explorar completamente de forma exata. A busca tabu é um aprimoramento das técnicas de busca local, introduzindo um mecanismo que controla e regula as soluções já visitadas para garantir uma busca mais abrangente e menos propensa a se estagnar em soluções subótimas. Nesta técnica o algoritmo começa com uma solução inicial e, a cada iteração, move-se para uma solução vizinha que é considerada uma melhoria (ou, em alguns casos, uma solução que não é necessariamente melhor, mas aceitável). Ao contrário de outras formas de busca local, a busca tabu possui um histórico que impede que o algoritmo volte a soluções já visitadas recentemente, ou que se mova para soluções que foram visitadas de forma recorrente. Esse mecanismo de memória é denominado lista tabu. A lista tabu armazena um conjunto de movimentos ou soluções já exploradas, e seu principal objetivo é evitar que o algoritmo explore essas soluções novamente por um período determinado, o que seria chamado de tabu. Em outras palavras, a lista tabu impede que o algoritmo faça movimentos cíclicos ou reanalise as mesmas soluções, promovendo uma exploração mais diversificada do espaço de soluções.

Em uma busca tabu o processo começa com uma solução inicial, que pode ser gerada aleatoriamente ou por meio de uma heurística simples. A partir dessa solução, o algoritmo começa a explorar as soluções vizinhas. À medida que o algoritmo realiza movimentos e explora novas soluções, ele registra certos movimentos (ou características das soluções) na lista tabu. Essas informações são armazenadas por um número limitado de iterações, após o qual podem ser removidas. O algoritmo avalia as soluções vizinhas com base em um critério de melhoria (por exemplo, minimização de custo ou maximização de ganho). Se uma solução vizinha for melhor que a solução atual e não estiver na lista tabu, o algoritmo a adota como a nova solução corrente. A lista tabu impede movimentos que envolvam soluções ou mudanças que já ocorreram recentemente. Caso não haja uma solução vizinha que atenda ao critério de melhoria sem ser tabu, o algoritmo pode ser forçado a aceitar uma solução pior temporariamente (ou seja, um movimento não melhor), como uma estratégia para escapar de ótimos locais. O algoritmo pode parar após um número fixo de iterações ou quando atingir um critério de qualidade de solução satisfatório. O quadro 1 apresenta algumas vantagens e desvantagens do uso desta técnica.

Quadro 1: Vantagens e desvantagens da busca tabu

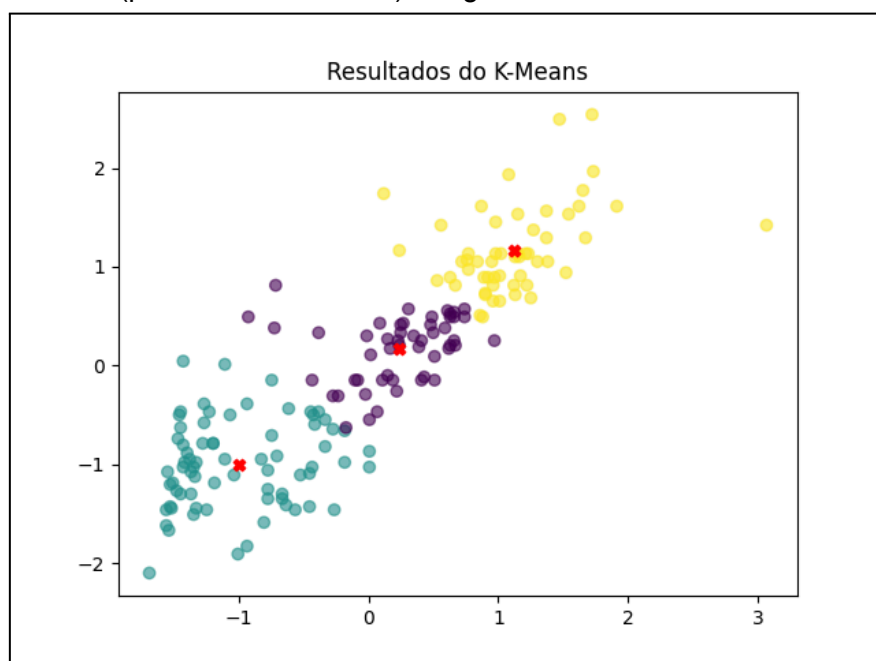
Vantagens	Desvantagens
<p>Evitar ótimos locais: Ao manter o controle sobre soluções já visitadas, a busca tabu evita que o algoritmo fique preso em soluções subótimas, permitindo que ele continue explorando o espaço de soluções.</p> <p>Exploração mais ampla: A lista tabu força a busca a explorar regiões novas do espaço de soluções, o que pode ser crucial em problemas complexos.</p> <p>Flexibilidade: A busca tabu é altamente personalizável, podendo ser ajustada para diferentes tipos de problemas, critérios de melhoria e estruturas de vizinhança.</p>	<p>Complexidade computacional: O processo de manter e atualizar a lista tabu pode ser custoso, principalmente em problemas com muitas soluções vizinhas e quando o espaço de busca é muito grande.</p> <p>Necessidade de parâmetros: A eficácia da busca tabu depende fortemente da escolha de parâmetros, como o tamanho da lista tabu, o número máximo de iterações e os critérios de aspiração. A escolha inadequada desses parâmetros pode afetar negativamente o desempenho do algoritmo.</p>

Fonte: Autores

Proposta das Vizinhanças

Na investigação de busca local deste trabalho foram propostas dois tipos de vizinhanças a partir dos centroides gerados pelo método de K-means. Na figura 1 são apresentados os centroides gerados pelo K Means para a base de dados reduzida (usando apenas os parâmetros “flavonoides” e “Fenóis Totais” como descritores dos registros). Estes centroides definem a referência de avaliação da performance dos algoritmos investigados neste trabalho.

Figura 1: centroides (pontos em vermelho) e registros da base reduzida de dados de vinho



Fonte: autores

Vizinhança I

Nesta primeira proposta de vizinhança, denominada de Vizinhança com Passo Simples (VPS), para cada um dos três centroides da figura 1 foram gerados um conjunto de 8 vizinhos a partir do incremento/ decrementos de um valor δ às suas coordenadas (x,y) . Nesta abordagem o valor de delta é incrementado/decrementado das coordenadas originais do centroide de forma tal que apenas uma única camada de vizinhos seja gerada. A figura 2 apresenta as coordenadas de cada um dos oito vizinhos de um centroide de coordenada (x,y) gerados pela abordagem da vizinhança VPS.

Figura 2: Vizinhos de um centroide de coordenada (x,y) na vizinhança VPS.

$(x-\delta, y+\delta)$	$(x, y+\delta)$	$(x+\delta, y+\delta)$
$(x-\delta, y)$	(x, y)	$(x+\delta, y)$
$(x-\delta, y-\delta)$	$(x, y-\delta)$	$(x+\delta, y-\delta)$

Visto que o problema de agrupamento dos registros de vinho trabalha simultaneamente com 3 centroides. Os oito vizinhos de cada um dos centroides da figura 1 devem ser combinados entre si para gerar o espaço de avaliação da vizinhança VPS. Dessa forma, a combinação dos vizinhos cria um espaço amostral de 512 possibilidades (8^3) de combinação de centroides.

Vizinhança II

Nesta segunda proposta de vizinhança, denominada de Vizinhança com Passo Duplo (VPD), para cada um dos três centroides da figura 1 foram gerados um conjunto de 24 vizinhos a partir do incremento/ decrementos de um valor δ às suas coordenadas (x,y) . Nesta abordagem o valor de delta é incrementado/decrementado das coordenadas originais do centroide de forma tal que apenas duas camadas de vizinhos sejam geradas. A figura 3 apresenta as coordenadas de cada um dos vinte e quatro vizinhos de um centroide de coordenada (x,y) gerados pela abordagem da vizinhança VPD.

Figura 3: Vizinhos de um centroide de coordenada (x,y) na vizinhança VPD.

$(x-2\delta, y+2\delta)$	$(x-\delta, y+2\delta)$	$(x, y+2\delta)$	$(x+\delta, y+2\delta)$	$(x+2\delta, y+2\delta)$
$(x-2\delta, y+\delta)$	$(x-\delta, y+\delta)$	$(x, y+\delta)$	$(x+\delta, y+\delta)$	$(x+2\delta, y+\delta)$
$(x-2\delta, y)$	$(x-\delta, y)$	(x, y)	$(x+\delta, y)$	$(x+2\delta, y)$
$(x-2\delta, y-\delta)$	$(x-\delta, y-\delta)$	$(x, y-\delta)$	$(x+\delta, y-\delta)$	$(x+2\delta, y-\delta)$
$(x-2\delta, y-2\delta)$	$(x-\delta, y-2\delta)$	$(x, y-2\delta)$	$(x+\delta, y-2\delta)$	$(x+2\delta, y-2\delta)$

Como no caso anterior, os vinte e quatro vizinhos de cada um dos centroides na abordagem VPD devem também ser combinados entre si para gerar o espaço de avaliação desta

vizinhança. Assim, a combinação dos vizinhos na abordagem VPD cria um espaço amostral de 13824 possibilidades (24^3) de combinação de centroides.

Resultados e Discussão

Aqui são apresentados os resultados obtidos nas seguintes investigações:

- Vizinhança I (512 possibilidades de combinação de centroides)
 - Busca local com vizinhança VPS - Primeira Melhora (VPS-PM)
 - Busca local com vizinhança VPS - Melhor Melhora (VPS-MM)
 - Busca Tabu com vizinhança VPS (Tabu-VPS)
- Vizinhança II: VPD (13824 possibilidades de combinação de centroides)
 - Busca Local com vizinhança VPD - Primeira Melhora (VPD-PM)
 - Busca Local com vizinhança VPD - Melhor Melhora (VPD-MM)
 - Busca Tabu com vizinhança VPD (Tabu-VPD)

Todos os resultados obtidos nas investigações previamente listadas são comparados com o resultado do K means original, apresentado na figura 1. A função objetivo que quantifica a qualidade de cada solução é o somatório das distâncias euclidianas dos pontos da base de dados em relação ao conjunto de centroides investigados. A busca tabu implementada para esta investigação usou 150 ciclos de busca e teve uma lista tabu com tamanho máximo (*tabu tenure*) de 150 registros.

Valores de referência gerados pelo K means

Na tabela 1 são apresentados os valores das coordenadas e do custo (função objetivo) dos centroides gerados pelo K means, apresentados na figura 1.

Tabela 1: Valores de referência para avaliação dos resultados das investigações

Método	centroides	Custo
k means (Ref)	[0.23242425 , 0,16964401] [-1.00234577 , -0,99855091] [1.11694924 , 1,17118187]	93,3534

Resultados das vizinhanças investigadas

A seguir para cada uma das vizinhanças investigadas é apresentado um resumo dos resultados obtidos. Para cada vizinhança é apresentada uma tabela e um conjunto de 3 gráficos. A tabela reúne os resultados (coordenadas dos centroides e custo) da primeira e melhor melhora, na busca local, bem como o resultado para a busca tabu. Os gráficos mostram o comportamento do custo ao longo da busca local e tabu.

Vizinhança I: VPS

Tabela 2: Valores obtidos na investigação da vizinhança I (VPS com 512 possibilidades)

Método	centroides	Custo
k means (Ref)	[0.23242425 , 0,16964401] [-1.00234577 , -0,99855091] [1.11694924 , 1,17118187]	93.3534
VPS-PM	[0.23242425 , 0,26966440] [-1.00234577 , -1,09855091] [1.11694924 , 1,07118187]	93.308
VPS-MM	[0.33242425 , 0,26964401] [-1.10234577 , -0,99855091] [1.11694924 , 1,07118187]	92.3988
Tabu-VPS	[0.13242425 , 0,03035599] [-1.20234577 , -1,09855091] [1.01694924 , 0,97118187]	92.274

Figura 4: Comportamento da função custo na busca local da vizinhança (VPS).

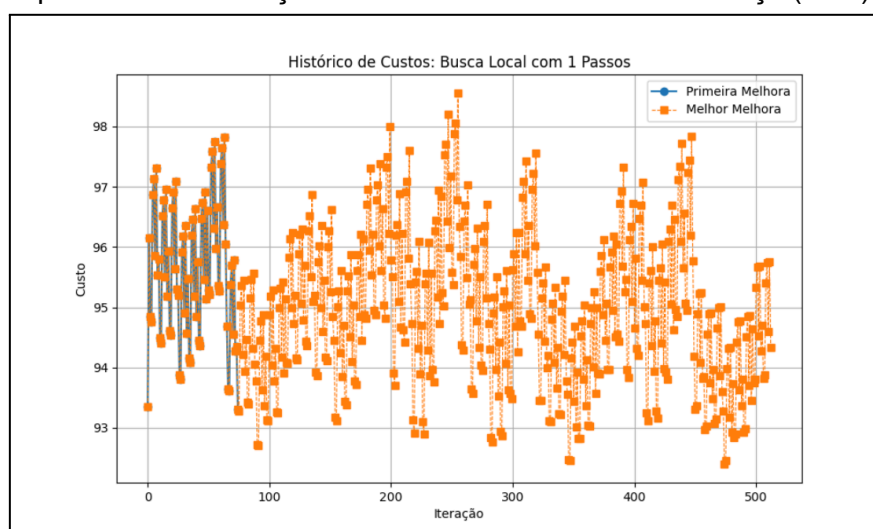


Figura 5: Comportamento da função custo na busca local da vizinhança (VPS), com zoom na região de encontro da primeira melhora.

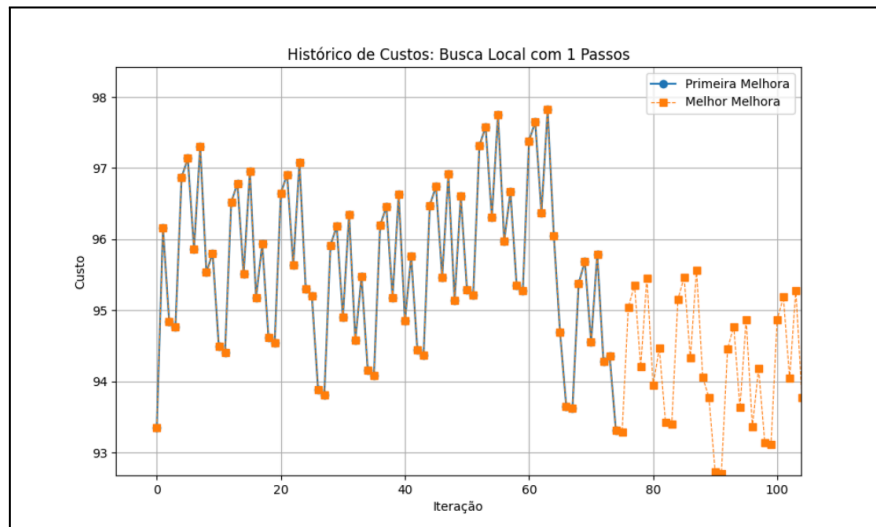
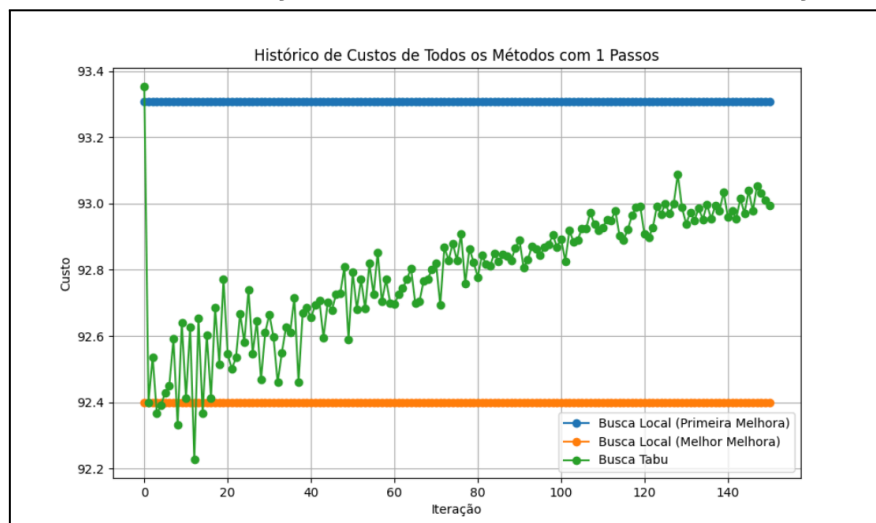


Figura 6: Comportamento da função custo na busca tabu com a vizinhança VPS.



Vizinhança II: VPD

Tabela 3: Valores obtidos na investigação da vizinhança II (VPD com 13824 possibilidades)

Método	centroides	Custo
k means (Ref)	[0.23242425 , 0,16964401] [-1.00234577 , -0,99855091] [1.11694924 , 1,17118187]	93.3534
VPS-PM	[0.33242425 , 0,36966440] [-1.10234577 , -0,99855091] [1.11694924 , 1,07118187]	93.2405
VPS-MM	[0.13242425 , -0,03035599] [-1.20234577 , -1,09855091] [1.01694924 , 0,97118187]	92.2274
Tabu-VPS	[0.13242425 , -0,03035599] [-1.20234577 , -1,09855091] [1.01694924 , 0,97118187]	92.274

Figura 7: Comportamento da função custo na busca local da vizinhança (VPd).

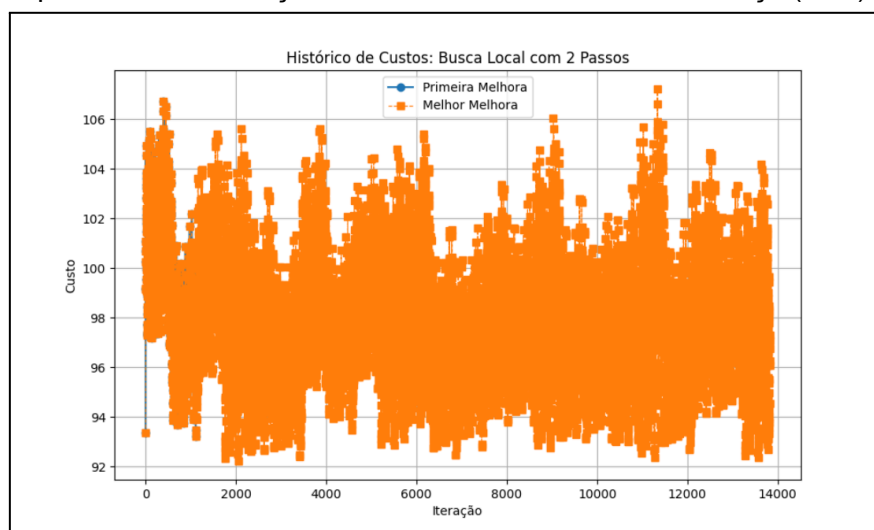


Figura 8: Comportamento da função custo na busca local da vizinhança (VPD), com zoom na região de encontro da primeira melhora.

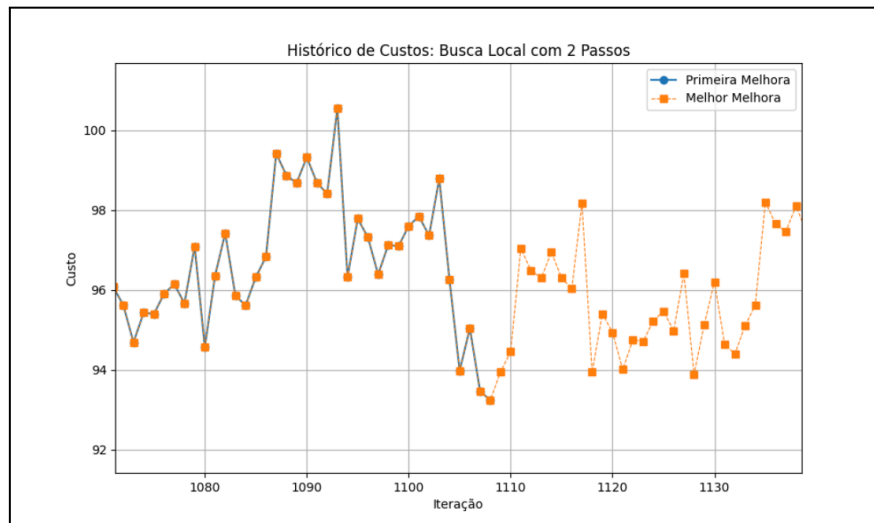
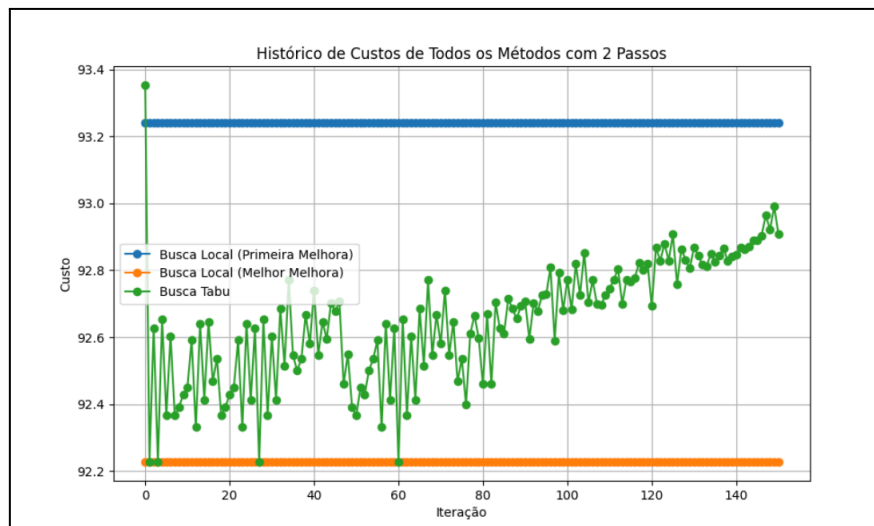


Figura 9: Comportamento da função custo na busca tabu com a vizinhança VPS.



Discussão dos resultados

Um resumo das informações apresentadas nas duas seções anteriores, com resultados separados para cada vizinhança, é apresentado na tabela 3. Nesta tabela verifica-se que para ambas as vizinhanças os resultados encontrados foram ao encontro das expectativas de aplicação das técnicas de otimização. Em ambas as vizinhanças a busca local, conseguiu melhorar os valores da função custo. Tanto na primeira melhora quanto na melhor melhora a vizinhança VPD apresentou melhor performance em comparação com a vizinhança VPS. A busca tabu no caso da vizinhança VPS foi a técnica que melhor performou, conseguindo reduzir o valor de referência da função custo em 1,2062%. Já no caso da vizinhança VPD verifica-se que a busca tabu reproduziu o resultado da melhor melhora na busca local da mesma vizinhança.

Tabela 3: Comparação dos valores da função custo para as duas vizinhanças investigadas para os casos: (i) De referência “K means”, (ii) busca local - primeira Melhora, (iii) busca local - Melhor melhora, (iv) busca tabu

Método	Custos VPS	Custos VPD
k means (Referência)	93.3534	
Primeira Melhora	93.308 <i>[Decréscimo de 0.0486%]</i>	93.2405 <i>[Decréscimo de 0.1209%]</i>
Melhor Melhora	92.3988 <i>[Decréscimo de 1.0226%]</i>	92.2274 <i>[Decréscimo de 1.2062%]</i>
Tabu	92.274 <i>[Decréscimo de 1.2062%]</i>	92.274 <i>[Decréscimo de 1.2062%]</i>

Nas avaliações de busca local, a ordem de investigação dos centroides do espaço amostral de combinações era a mesma, tanto no caso de primeira melhora como também no caso de melhor melhora. As figuras 5 e 8 apresentaram, para a vizinhança VPS e VPD, respectivamente, o intervalo de iterações no qual a primeira melhora foi encontrada. No caso da vizinhança VPS a primeira melhora foi encontrada no intervalo [70,80] das 512 possíveis interações, enquanto que para a vizinhança VPD a primeira melhora é encontrada no intervalo [1100,1110] das 13824 possíveis interações. As figuras 6 e 9 apresentam o comportamento da busca tabu para a vizinhança VPS e VPD, respectivamente. Em ambos os casos verifica-se uma redução expressiva da função custo já nos primeiros ciclos de iteração, com gradativa perda de qualidade de solução (aumento da função custo) a longo prazo.

Considerações finais

Este trabalho investigou uso de técnicas de busca local, como a primeira melhora, melhor melhora e busca tabu, aplicadas a uma base de dados de dimensionalidade reduzida gerada a partir da base "Wine Quality". As técnicas de busca local demonstraram ser eficientes para explorar o espaço de soluções de uma vizinhança específica, com a primeira melhora e a melhor melhora oferecendo soluções rápidas, mas com limitações em relação à qualidade das soluções globais. A busca tabu, por sua vez, mostrou um desempenho superior em termos de evitar soluções subótimas, apresentando uma capacidade de escapar de mínimos locais ao longo do processo de busca, embora com maior custo computacional. Esses resultados indicam que, em problemas de classificação como o analisado, a busca tabu pode ser mais vantajosa quando há necessidade de explorar mais profundamente o espaço de soluções, embora um equilíbrio entre desempenho e custo computacional deva ser cuidadosamente considerado.