



UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE EDUCAÇÃO SUPERIOR DO ALTO VALE DO ITAJAÍ –
CEAVI DEPARTAMENTO DE ENGENHARIA DE SOFTWARE - DESO

Breno José Coelho (breno.jose.coelho@gmail.com)

ALGORITMOS HEURÍSTICO BUSCA LOCAL ITERADA (iterated local search – ILS)

1. Introdução

Uma indústria farmacêutica está desenvolvendo um novo medicamento para tratar uma doença específica e, para isso, precisa selecionar os componentes químicos mais adequados. Esses componentes estão organizados em bancos de dados fornecidos por diferentes fornecedores, com informações sobre possíveis interações entre pares de componentes.

Essas interações são expressas em termos de intensidade: valores positivos indicam benefícios, como maior eficácia no combate à doença, enquanto valores negativos representam prejuízos, como a produção de efeitos colaterais indesejados. Quando a interação entre dois componentes não é informada, presume-se que ela seja neutra, ou seja, o impacto conjunto é considerado zero.

O problema em questão exige a escolha de um subconjunto de componentes que maximize os benefícios totais das interações, considerando as restrições de compatibilidade entre os componentes. Para isso, foi selecionada a **Busca Local Iterada (ILS)**, uma metaheurística capaz de explorar e refinar soluções de forma eficiente, escapando de mínimos locais para alcançar melhores resultados globais.

2. Representação da Solução

A solução é representada como um conjunto de índices que correspondem aos componentes químicos selecionados. Esse conjunto é manipulado ao longo do algoritmo, com operações de inclusão ou remoção de componentes para explorar diferentes combinações possíveis.

A matriz esparsa fornecida no banco de dados representa as interações entre os componentes. Cada elemento da matriz indica o grau de interação entre dois componentes, onde

valores positivos representam benefícios e valores negativos indicam prejuízos.

3. Estratégias Utilizadas

Foi adotado uma estratégia de Construção no qual a solução inicial é gerada de forma aleatória, selecionando um subconjunto de 50 componentes entre os disponíveis no banco de dados. Essa abordagem inicializa o processo de otimização com uma solução viável, mas potencialmente subótima, permitindo que o algoritmo explore melhorias subsequentes.

Para a avaliação da solução temos que o valor de uma solução é calculado somando todas as interações entre os componentes selecionados. Isso é feito extraíndo uma submatriz da matriz esparsa correspondente aos índices selecionados e somando seus elementos.

3.1 Estratégias de Busca Local

Bom para o ILS usamos a estratégia da vizinhança, onde foi feito para que a vizinhança de uma solução seja gerada ao incluir ou excluir um único componente químico do conjunto atual. Essa operação cria uma nova solução candidata que é avaliada para verificar se representa uma melhoria em relação à solução atual.

Assim a busca local percorre as soluções na vizinhança da solução atual, aceitando uma solução vizinha apenas se ela melhorar o valor da solução. Caso uma solução melhor seja encontrada, ela se torna a nova solução atual, e o processo continua até que não sejam identificadas mais melhorias.

3.2 Estratégia de Perturbação

A perturbação é aplicada para escapar de mínimos locais e explorar regiões diferentes do espaço de soluções. Durante a perturbação, até 60 componentes do subconjunto atual são adicionados ou removidos de forma aleatória. Isso cria uma nova solução de partida para o processo de busca local.

4. Algoritmo Geral (ILS)

A função `ils` (Iterated Local Search) utiliza uma combinação de busca local e perturbação controlada para otimizar soluções em problemas combinatórios. Inicia-se com uma solução aleatória, selecionando 50 componentes de um conjunto de dados. Essa solução é avaliada com a função `evaluate_solution`, que calcula o valor total da interação entre os componentes.

A cada iteração, a solução passa por uma busca local para refinar os componentes, buscando a melhor alteração. Se a nova solução for melhor que a anterior, ela se torna a melhor solução global. Para evitar mínimos locais, é aplicada uma etapa de perturbação, onde componentes são aleatoriamente removidos ou adicionados à solução. Após a perturbação, a busca local é novamente aplicada. Esse ciclo se repete até o número máximo de iterações.

Ao final, a função retorna a melhor solução encontrada, representando os componentes que maximizam a interação química conforme os critérios definidos.

5. RESULTADOS

Foram utilizados 10 bancos de dados fornecidos, cada um contendo informações sobre 500 componentes químicos. Cada instância foi processada duas vezes para obter média e desvio padrão dos resultados.

Métricas Avaliadas:

Melhor Valor: A maior soma de interações positivas obtida.

Valor Médio: A média dos valores obtidos nas replicações.

Tempo Médio de Execução: O tempo médio necessário para cada execução.

Instância	Melhor valor	Valor médio	Tempo médio
1	60672	60668.00	405.45 /s
2	58258	58239.50	399.15 /s
3	55376	55369.00	405.53 /s
4	50554	50553.50	392.49 /s
5	61830	61830.00	383.33 /s
6	58205	58192.50	405.35 /s
7	62728	62727.00	399.77 /s
8	50452	50437.50	402.77 /s
9	56198	56180.50	405.07 /s
10	57343	57343.00	393.51 /s

6. CONCLUSÃO

O algoritmo ILS se mostrou eficiente na exploração do espaço de soluções, conseguindo identificar valores elevados para as interações totais. A combinação entre busca local e perturbação desempenhou um papel crucial, permitindo que o algoritmo escapasse de mínimos locais e encontrasse soluções de alta qualidade. A adoção de uma matriz esparsa contribuiu significativamente para a redução da complexidade computacional, o que possibilitou a análise de instâncias maiores dentro de um tempo de execução razoável. Além disso, o desempenho do algoritmo foi consistente nas diferentes execuções, com os valores médios obtidos ficando próximos aos melhores valores encontrados, o que demonstra sua robustez.

Porém, uma limitação observada foi a escolha aleatória da solução inicial, que pode influenciar o tempo necessário para alcançar a convergência em algumas instâncias. Em futuras implementações, estratégias de inicialização mais informadas poderiam ser exploradas, a fim de otimizar ainda mais o desempenho do algoritmo.

REFERÊNCIAS

Repositório GitHub: <https://github.com/BrenoJoseCoelho/AlgoritimoLS>