Data:16/02/2024

RELATÓRIO TRABALHO PRÁTICO DE OTIMIZAÇÃO

A. INTEGRANTES

Nome Completo: Luis Filipe Antunes Rodrigues (00314848)
Email: lufi96@hotmail.com

Nome Completo: Bibiana Duarte

B. RESUMO

Neste trabalho realizamos a implementação e avaliação de heurísticas para o problema de otimização Viajens Felizes. Começamos descrevendo o problema e suas restrições, como foi realizada a implementação utilizando um solver genérico, a implementação da heurística baseada no algoritmo de Simulated Annealing e por fim mostramos os resultados obtidos

C. PROBLEMA

Uma **instância** do problema Viajens Felizes consiste em um conjunto de n pessoas que querem viajar. Cada uma é disposta a pagar um valor vi (em R$), i ∈ [m] para participar numa viagem e tem um peso de pi (em kg). O problema é que tem somente m aviões com capacidades ci, i ∈ [m] (em kg). Além disso, existem relações de amizade entre as pessoas. Isso se manifesta da seguinte forma: caso pessoas i, j viajam no mesmo avião, eles pagam um valor vij (em R$) a mais (os valores são simétricos, i.e. vij = vji).

Uma **solução** consiste em uma seleção de pessoas que vão viajar, junto com uma alocação das pessoas aos aviões, respeitando as capacidades dos aviões.

O **objetivo** do problema é maximizar o valor total, i.e. o valor vi recebido por cada pessoa i ∈ [n] selecionada, além dos valores vij recebidos por pessoas viajando no mesmo avião

As restrições para o problema são as seguintes:

Formulação:

$$\text{maximize} \sum_{i}^{n}\sum_{a}^{m} t_{ai} \cdot V_i + \sum_{i}^{n}\sum_{j}^{n}\sum_{a}^{m} T_{aij} \cdot V_{ij}$$

$$\text{sujeito a} \quad \sum_{i}^{n} t_{ai} \cdot P_i \leq C_a$$

$$\sum_{a}^{m} t_{ai} \leq 1 \qquad \forall i \in [n]$$

$$T_{aij} \leq \frac{t_{ai}+t_{aj}}{2} \quad \forall i \in [n], \forall j \in [n], \forall a \in [m]$$

- P_i é o peso da pessoa i.
- V_i é o valor que a pessoa i pagaria para viajar
- V_{ij} é o valor a mais que a pessoa i pagaria para viajar com a pessoa j.
- t_{ai} indica se a pessoa i está viajando ou não no avião a, caso esteja o valor será 1, e 0 caso contrário
- T_{aij} indica se duas pessoas i e j estão viajando ou não em um avião a caso estejam o valor será 1, e 0 caso contrário

D. SOLVER

Para implementar o solver utilizamos linguagem Julia com pacote de solver GLPK. As restrições adicionadas seguem o modelo descrito anteriormente

E. HEURÍSTICA

Realizamos a avaliação de **duas heurísticas**. A primeira é baseada na ideia de **trocar pessoas entre aviões (H1)**, pois permite uma **otimização mais local** da solução atual, e a segunda **selecionar pessoas ainda não alocadas a aviões (H2)** e substituí-las por pessoas já alocadas, que permite uma **otimização mais global**, pois exploramos ainda mais o espaço de soluções. Mais detalhadamente, tentamos selecionar alguma pessoa ainda não alocada que agregue maior valor ao avião selecionado e substituí-la por alguma pessoa que tenha peso maior ou igual ao seu, pois assim o avião ainda pode comportá-la.

Para a criação de uma **solução inicial,** inicialmente tentamos alocar as pessoas baseada no valor. Entretanto, notamos que realizando **ordenamento das pessoas pela razão valor/peso** e alocando nos aviões até que atinja a capacidade máxima (semelhante ao problema da mochila) gera soluções iniciais de maior valor.

E. SIMULATED ANNEALING

Para a exploração do espaço de soluções, utilizamos o algoritmo de Simulated Annealing. Resumidamente, o algoritmo de simulação de recozimento (simulated annealing) é uma técnica de otimização inspirada no processo de recozimento do metal na metalurgia. Ele funciona explorando soluções de forma probabilística, permitindo movimentos para soluções piores inicialmente, com a esperança de encontrar o ótimo global. Durante o

processo, a temperatura é gradualmente reduzida, o que controla a probabilidade de aceitar soluções piores à medida que a busca avança, permitindo que o algoritmo escape de mínimos locais.

As entradas para o algoritmo são:

- Temperatura inicial (T0): Define a temperatura inicial do sistema. Geralmente, é escolhida alta o suficiente para permitir uma exploração ampla do espaço de busca.
- Taxa de resfriamento (alpha): Determina a taxa na qual a temperatura diminui ao longo do tempo. Valores típicos estão na faixa de 0,8 a 0,99.
- Número de iterações por temperatura: Define quantas iterações são realizadas em cada temperatura antes de diminuí-la.
- Critério de parada: Determina a condição para encerrar a busca, como um número máximo de iterações ou uma melhoria mínima no valor da função objetivo.

E. RESULTADOS

Para avaliar os resultados da heurística vamos fixa número de aviões m = 10 cada um com capacidade 0.8/m do valor da soma dos pesos de todas pessoas da instância.

Com a heurística H1:

Parâmetros T0 = 1000, Tf = 0.01, r = 0.95, Tlength = 1000

Instância	Valor da solução inicial	Valor da melhor solução encontrada	Melhor valor conhecido
vf01	5622	6259	10385
vf02	5194	6476	10413
vf03	5371	6434	10608
vf04	4883	6664	11066
vf05	5629	6660	10702

Com a heurística H2

Não computada em tempo polinomial, são necessárias muitas iterações para calcular custos associados a à substituição de uma pessoa por outra de um avião

Para o solver não conseguimos implementar um sistema satisfazível.

F. CONCLUSÕES

A escolha de uma heurística impacta significativamente no desempenho de um algortimo.

G. LINK PARA O CÓDIGO

https://github.com/lfarodrigues/trabalho_otimizacao