**UENF**

Universidade Estadual do Norte Fluminense Darcy Ribeiro

**Curso:** Ciência de Computação **Data:** 20./.10./2022

Trabalho: Trabalho2 Período: 7º Disciplina: Top. Esp.: Heurísticas e Complexidade

Professor: Fermín Alfredo Tang Turno: Diurno

Nome do Aluno: João Bosco Pedrosa Filho Matrícula: 00119110351

1.- Com base nos problemas estudados no Trabalho1, escolher um problema da categoria NP da sua preferência e realize as seguintes tarefas:

Obs: O problema da categoria NP escolhido foi o problema da mochila.

i) Propor e implementar uma heurística construtiva para achar uma solução inicial para o problema escolhido. Descrever a representação da sua solução e seu método construtivo e implemente.

Resposta: A proposição é de um problema em que haverá uma bolsa de mercado que pode armazenar determinada quantidade de doces em um mercado, ao total serão analisados um conjunto de 10 doces diferentes, em que 3 doces serão escolhidos de forma aleatória(heurística de construção aleatória) em um conjunto de 3 testes no total. A melhor solução encontrada nesse conjunto de testes será a solução aceita. A representação pode ser feita por meio de uma tabela na qual serão armazenadas as informações dos doces, como peso, benefício e a informação da capacidade da bolsa.

Abaixo será vista a tabela que será usada para testar as heurísticas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Doce** | **Peso(mg)** | **Custo/Benefício** | **Benefício/Peso** | **Capacidade da Bolsa** |
| Jujuba |  |  |  |  |
| Maria-mole |  |  |  |  |
| Barra de chocolate |  |  |  |  |
| Paçoca |  |  |  |  |
| Bolinho |  |  |  |  |
| Pote de doce de leite |  |  |  |  |
| Geléia de Mocotó |  |  |  |  |
| Saco de balas de goma |  |  |  |  |
| Pote de pasta de amendoim |  |  |  |  |
| Trufa pequena |  |  |  |  |

**Implementação:** Abaixo poderá ser visto o código fonte referente ao problema descrito acima, além de uma explicação sobre como o código se comporta, suas variáveis, métodos. Além disso, serão mostrados prints de janelas de execução, para análise de como o código roda.

**Código Fonte:**

#Doces

import random

capacidade\_bolsa = 600 # A capacidade máxima da bolsa

doces = {  #A chave indica o nome do doce, o primeiro indice da lista indica o peso do doce e o segundo indica o benefício

    'Jujuba': [40, 1],

    'Maria-mole': [70, 2],

    'Chocolate': [250, 6],

    'Paçoca': [35, 3],

    'Bolinho': [120, 2],

    'Doce de Leite':[350, 4],

    'Geleia de Mocotó': [170, 2],

    'Saco de Balas de Goma': [430, 3],

    'Pote de Pasta de Amendoim': [400, 4],

    'Trufa': [30, 2]

}

def Doce\_Aleatorio():

    doce = random.choice(list(doces.keys())) #escolhe um doce aleatório, key = 'Doce qualquer'

    return doce, doces[doce][0], doces[doce][1] #retorna o doce, o seu peso e o seu benefício

def Teste():

    doce\_1, peso\_doce\_1, beneficio\_doce\_1 = Doce\_Aleatorio() #Seleção de um doce aleatório

    doce\_2, peso\_doce\_2, beneficio\_doce\_2 = Doce\_Aleatorio() #Seleção de outro doce aleatório

    doce\_3, peso\_doce\_3, beneficio\_doce\_3 = Doce\_Aleatorio() #Seleção de outro doce aleatório

    Pesos\_Doces = [peso\_doce\_1, peso\_doce\_2, peso\_doce\_3] #Função para guardar somente o peso dos doces

    Beneficio\_Doces = [beneficio\_doce\_1, beneficio\_doce\_2, beneficio\_doce\_3] #Função para guardar somente o benefício dos doces

    print(f'Os doces escolhidos foram {doce\_1}, {doce\_2} e {doce\_3}')

    soma\_pesos = sum(Pesos\_Doces) #soma total do peso dos doces escolhidos

    soma\_beneficios = sum(Beneficio\_Doces) #soma total do benefício dos doces escolhidos

    print(f'A soma total do peso dos doces é {soma\_pesos}')

    print(f'A soma total do benefício da compra é {soma\_beneficios}')

    Solução = [soma\_pesos, soma\_beneficios] #A solução

    print(Solução)

    if (soma\_pesos > capacidade\_bolsa):

        print("Ultrapassou a capacidade da bolsa")

    else:

        print("É uma solução válida, pois a soma dos pesos dos doces não ultrapassa a capacidade máxima da bolsa ")

    print("\n")

    return Solução

def Filtro\_Soluções(Solucoes): #Função para encontrar a melhor solução

    capacidade\_bolsa = 600

    melhor\_solução = [-1, -1]

    for elemen in Solucoes: # percorre cada solução

        if (elemen[0] > capacidade\_bolsa): #peso > capacidade não é permitido, solução ignorada

            next

        else: #solução válida

            if (elemen[1] > melhor\_solução[1]): #benefício maior que a melhor solução

                melhor\_solução = elemen #nova melhor solução

            if(elemen[0] == melhor\_solução[0] and elemen[1] > melhor\_solução[0]):#Se tem o mesmo peso que a melhor solução e beneficio maior

                melhor\_solução = elemen #nova melhor solução

    print(f'A melhor solução é {melhor\_solução}')

S1 = Teste() #chamada de função

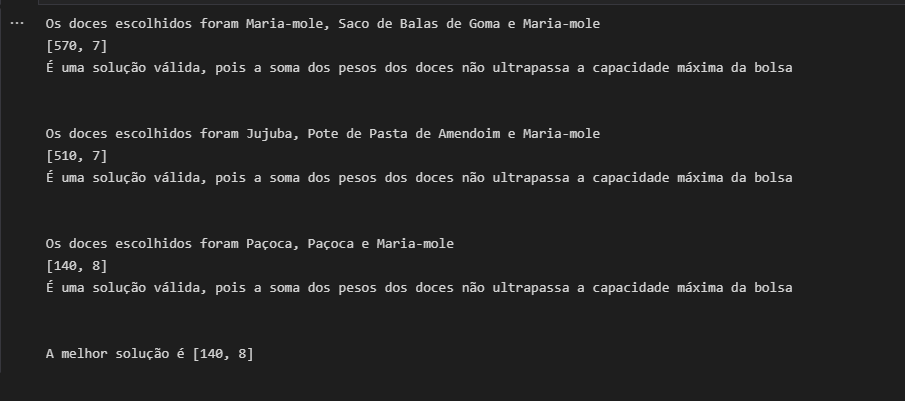
S2 = Teste()

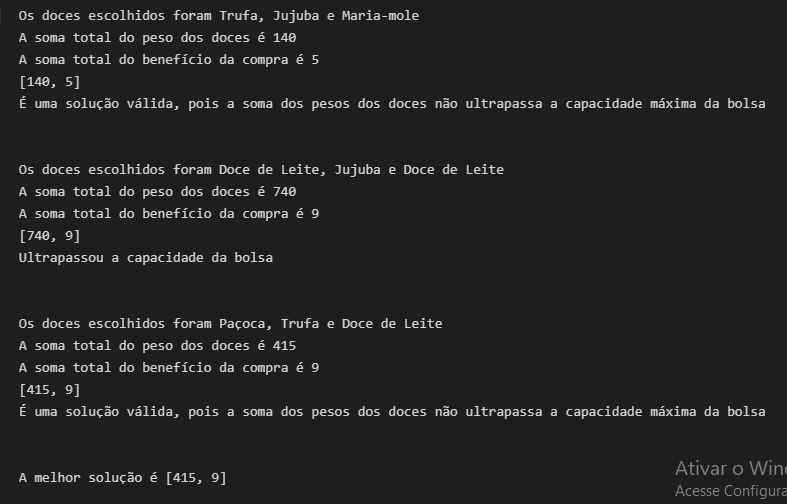
S3 = Teste()

Solucoes = [S1, S2, S3] #junção das soluções

Filtro\_Soluções(Solucoes) # filtragem da melhor solução

Além do código, a seguir serão mostradas duas janelas de execução(o que também se enquadra para o item 3, que pede testes para instâncias pequenas do problema, porém optei por fazer testes de mesa também).

Janela de execução 1



Janela de execução 2

ii) Propor e implementar uma heurística de busca local para achar uma solução de melhor qualidade que a encontrada no item i). Descrever o mecanismo de busca local adotado e implemente.

Resposta: A proposta é que na heurística de busca local, sejam selecionados os mesmos doces com os mesmos valores da questão acima, mas o que mudará será que justamente a forma com que será encontrada uma solução. O primeiro passo será definir a solução inicial, que será justamente a solução encontrada no teste do item iii. Após isso serão feitas comparações entre as soluções para garantir que houve uma melhora de qualidade. A estratégia será pegar essa solução inicial e realizar uma pequena mudança na mesma, trocando um doce da sua sacola, selecionando um novo doce disponível que não estava presente nessa solução e depois comparando a nova solução com a inicial, o objetivo é conseguir uma melhora e o limite de testes é 5. Será utilizada a estratégia de primeiro aprimorante, em que a combinação de soluções do conjunto de doces será explorada até encontrar uma primeira solução que seja melhor que a solução inicial. Caso não seja encontrada solução melhor, será considerada que a solução inicial é um ótimo local.

Implementação: Infelizmente não consegui implementar a tempo pois já está perto do fim do prazo. Mas a ideia que passa pela minha mente seria de pegar o código que fiz acima e continuá-lo, pegando a melhor solução, fazendo um random.choice que tiraria um dos doces selecionados aleatoriamente e depois chamaria novamente a função de Doce Aleatório, para chamar um doce pro lugar do retirado, e faria esse teste até que fosse achada uma solução melhor ou então chegasse no limite de testes.

iii) Teste as suas duas heurísticas para instâncias pequenas do problema.

Resposta(heurística construtiva):

A seguir poderá ser visto um teste de como funcionaria para a heurística construtiva.

O primeiro passo será reinserir a tabela vista acima, agora colocando valores para cada doce. A tabela poderá ser vista na página abaixo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Doce** | **Peso(mg)** | **Benefício** | **Benefício/Peso** | **Capacidade da Bolsa(600mg)** |
| Jujuba | 40 | 1 | 0,025 |  |
| Maria-mole | 70 | 2 | 0,028 |  |
| Barra de chocolate | 250 | 6 | 0,024 |  |
| Paçoca | 35 | 3 | 0,086 |  |
| Bolinho | 120 | 2 | 0,017 |  |
| Pote de doce de leite | 350 | 4 | 0,011 |  |
| Geléia de Mocotó | 170 | 2 | 0,012 |  |
| Saco de balas de goma | 430 | 3 | 0,007 |  |
| Pote de pasta de amendoim | 400 | 4 | 0,010 |  |
| Trufa | 30 | 2 | 0,070 |  |

Após isso, será feita a escolha aleatória de 3 doces por 3 rodadas e então será definida a solução para a heurística construtiva.

Exemplo: Rodada 1 → Pote de pasta de amendoim, barra de chocolate, paçoca.

Rodada 2 → Saco de balas de goma, trufa , bolinho.

Rodada 3 → Maria mole, bolinho, barra de chocolate.

Definidas as três soluções aleatórias, agora o próximo passo seria comparar dentre tais soluções, qual seria melhor.

Serão feitas tabelas para analisarmos os casos aleatórios escolhidos e qual solução escolher.

As tabelas para cada rodada podem ser vistas nas páginas abaixo.

Rodada 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Doce** | **Peso(mg)** | **Benefício** | **Benefício/Peso** | **Capacidade da Bolsa(600mg)** |
| Pote de pasta de amendoim | 400 | 4 | 0,010 |  |
| Barra de chocolate | 250 | 6 | 0,024 |  |
| Paçoca | 35 | 3 | 0,086 |  |
| Total | 685 |  |  | x |

Resposta da rodada 1: A primeira solução encontrada não é viável, pois ultrapassa a capacidade da bolsa.

Rodada 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Doce** | **Peso(mg)** | **Benefício** | **Benefício/Peso** | **Capacidade da Bolsa(600mg)** |
| Saco de balas de goma | 430 | 3 | 0,007 |  |
| Trufa | 30 | 2 | 0,070 |  |
| Bolinho | 120 | 2 | 0,017 |  |
| Total | 580 | 7 | 0,012 | 600 |

Resposta da rodada 2: A solução encontrada para essa rodada é válida para o problema, ela será armazenada até o fim dos testes e se necessário, comparada com outras soluções.

Rodada 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Doce** | **Peso(mg)** | **Benefício** | **Benefício/Peso** | **Capacidade da Bolsa(600mg)** |
| Maria-Mole | 70 | 2 | 0,028 |  |
| Bolinho | 120 | 2 | 0,017 |  |
| Barra de Chocolate | 250 | 6 | 0,024 |  |
| Total | 440 | 10 | 0,023 |  |

Resposta da rodada 3: A solução encontrada para essa rodada também é válida para o problema, já que o peso total não excedeu a capacidade disponível da bolsa.

Agora, chegamos em um ponto em que foram encontradas duas soluções para o problema, quando isso ocorre, a idéia é comparar as soluções aleatórias encontradas no domínio de soluções possíveis para o meu problema proposto e definir qual seria a melhor.

Solução da rodada 2: Peso total: 580, benefício: 7, benefício/Peso total: 0,012.

Solução da rodada 3: Peso total: 440, benefício:10, benefício/Peso total: 0,023.

Analisando ambas as soluções, pode-se determinar que a solução da terceira rodada de testes aleatórios trouxe a melhor solução para essa heurística construtiva proposta, já qiue o peso total foi menor e ainda assim teve mais benefícios para o cliente, como pode ser observado na razão entre o benefício e o peso total. Agora, o passo final é pegar a melhor solução encontrada no conjunto de testes e defini-la como solução do problema.

**Solução**: Rodada 3 → Maria mole, bolinho, barra de chocolate.

Resposta(Heurística de Busca Local): Agora será analisada a proposta que fiz para a heurística de busca local. Como dito anteriormente, o primeiro passo seria ter uma solução inicial, que será justamente a solução encontrada na Heurística Construtiva.

Solução inicial: Maria mole, bolinho, barra de chocolate.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Doce** | **Peso(mg)** | **Benefício** | **Benefício/Peso** | **Capacidade da Bolsa(600mg)** |
| Maria-Mole | 70 | 2 | 0,028 |  |
| Bolinho | 120 | 2 | 0,017 |  |
| Barra de Chocolate | 250 | 6 | 0,024 |  |
| Total | 440 | 10 | 0,023 |  |

Após isso, será retirado um desses doces e então será substituído por outro doce, feita essa substituição, haverá uma comparação entre as soluções, visando achar uma solução melhor que a inicial. Suponhamos que o doce retirado seja a maria mole, após isso, será escolhido um doce para substitui-la e então recalcularemos a nossa solução. Suponhamos que neste caso o doce escolhido foi a geleia de mocotó. Analisemos como ficaria a tabela abaixo.

Solução s1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Doce** | **Peso(mg)** | **Benefício** | **Benefício/Peso** | **Capacidade da Bolsa(600mg)** |
| Geleia de mocotó | 170 | 2 | 0,012 |  |
| Bolinho | 120 | 2 | 0,017 |  |
| Barra de Chocolate | 250 | 6 | 0,024 |  |
| Total | 540 | 10 | 0,018 | 600 |

Observe que a primeira solução encontrada(s1) não foi uma solução considerada melhor do que a solução inicial que já temos, já que o peso total foi maior e isso não implicou em um benefício maior para o cliente.

Após isso, como não foi achada uma solução melhor, será feito um novo teste. Nesse caso o doce substituído será o bolinho e o adicionado será a trufa.

Solução s2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Doce** | **Peso(mg)** | **Benefício** | **Benefício/Peso** | **Capacidade da Bolsa(600mg)** |
| Maria-Mole | 70 | 2 | 0,028 |  |
| Trufa | 30 | 2 | 0,070 |  |
| Barra de Chocolate | 250 | 6 | 0,024 |  |
| Total | 350 | 10 | 0,028 |  |

Observe como esta troca diminuiu o peso total da bolsa e ao mesmo tempo manteve o benefício para o cliente que está realizando a compra. Neste caso, a nova solução encontrada, s2, é melhor que a solução inicial, então haverá uma troca entre a solução inicial e s2 , com s2 se tornando a solução do problema.

iv) Teste as suas duas heurísticas para instâncias mais difíceis. Por exemplo, procure conjuntos de dados de teste (*test data sets*) para o problema escolhido, em repositórios semelhantes a: <http://people.brunel.ac.uk/~mastjjb/jeb/info.html> e escolha um conjunto de pelo menos três problemas para resolver. Mostre os resultados em uma tabela. Se possível compare a sua solução com a solução ótima, caso disponível.

Não consegui fazer a tempo.