

# Aplicação da *Tabu Search* no Problema MAX-KBQF

## Atividade 5

### Grupo 7

Ítalo Fernandes Gonçalves - RA: 234990

Lucas Guesser Targino da Silva - RA: 203534

Luiz Gustavo Silva Aguiar - RA: 240499

Maio de 2022

MC859/MO824

# 1 Definição do Problema

Neste atividade, vamos abordar o problema denominado por *Maximum Knapsack quadratic binary function* (MAX-KQBF). No MAX-KQBF temos uma mochila de capacidade  $W$  e um conjunto de itens  $I$ , onde cada item  $i \in I$  tem um peso  $w_i \in \mathbb{R}$ . Desejamos empacotar os itens  $I$  na mochila, de forma a maximizar uma função quadrática e respeitando a capacidade da mochila, isto é, a soma dos pesos dos itens colocados na mochila precisa ser menor ou igual a  $W$ . Este problema pode ser formulado da seguinte forma:

$$\text{Max} \quad \sum_{i=1}^n \sum_{j=1}^n a_{ij} \cdot x_i \cdot x_j \quad (1)$$

$$\text{Sujeito a} \quad \sum_{i=1}^n w_i \cdot x_i \leq W \quad (2)$$

$$x_i \in \{0, 1\} \quad \forall i = \{1, \dots, n\} \quad (3)$$

onde  $a_{ij}$ ,  $w_i$  e  $W \in \mathbb{R}$  são parâmetros do problema, e  $n = |I|$ . O MAX-KQBF é uma generalização do problema *Maximum Quadratic Binary Function* (MAXBQF), no qual é um problema NP-difícil [2]

## 2 Tabu Search

A metaheurística *Tabu Search* (TS) é um procedimento adaptativo auxiliar, que direciona um algoritmo de busca local na exploração contínua dentro de um espaço de busca. Diferente de outras abordagens, a TS evita que retornar a um ótimo local visitado previamente de forma a superar a otimalidade local [1].

Nessa abordagem, o princípio básico parte da ideia de direcionar a busca local, permitindo movimentos sem melhoria. O retorno as soluções visitadas previamente é evitado com uma memorização através das chamadas listas tabu. Essas listas armazenam o histórico recente da busca. Sua implementação consiste basicamente na definição de uma busca local e na estratégia de memorização das listas tabu. Tais etapas serão descritas sobre a ótica do problema MAX-KQBF, adentrando na adequação da abordagem ao problema.

## 2.1 Solução inicial

Utilizamos a heurística construtiva do método GRASP para encontrarmos uma solução inicial para o problema. A solução foi construída a partir de uma iteração do GRASP e ela foi construída de forma puramente gulosa tomando assim  $\alpha = 0$ .

## 2.2 Lista Tabu

A lista tabu que é uma das principais características dessa abordagem, evita que a busca seja cíclica, voltando em soluções previamente exploradas e ainda colaborando para a realização de uma busca mais densa dentro do espaço de busca. Cada passo da busca é registrado na lista segundo as regras de ativação. Depois de adicionados permanecem na lista por um determinado período de iterações controlado pelo parâmetro *tabu tenure*. Nas implementações realizadas, esse valor foi definido como uma razão do tamanho  $n$  da entrada, e foram utilizados os valores  $0.2 \cdot n$  e  $0.4 \cdot n$ .

## 2.3 Critério de aspiração

Os tabus, como são chamadas as memorizações realizadas, são bem decisivas pois podem proibir movimentos atraentes mesmo quando não há possibilidade de ciclos ou podem levar a estagnação geral no processo de busca. O critério de aspiração permite cancelar memorizações realizadas. Para a atual abordagem ao problema MAX-KQBF foi utilizado o critério de aspiração mais simples e utilizado, que consiste em permitir o movimento, mesmo que seja tabu, se resultar em uma solução com valor objetivo melhor que o da melhor solução conhecida.

## 2.4 Operadores de busca

A heurística de busca local com a estratégia *Best Improvement* consiste em analisar todas as soluções vizinhas e se mover para aquela que tiver a melhor avaliação e que represente uma melhora em relação à solução corrente. Se não houver solução de melhora, então o método para e retorna a solução corrente como ótimo local em relação à estrutura de vizinhança utilizada. Diferente do método anterior, a *First improvement*, a decisão para mover-se para uma

nova solução vizinha não ocorre após a análise de toda a vizinhança mas sim na primeira solução vizinha encontrada melhor que a solução atual.

## 2.5 Estratégias tabu alternativas

Foram utilizadas 2 estratégias tabu alternativas: *diversification by restart* e *intensification by restart*. A estratégia *Diversification by restart* constroi uma nova solução (aleatorizada) e recomeça o processo de busca. A ideia dela é explorar regiões não atingidas pelo caminho até então percorrido. Já a estratégia *intensification by restart* faz uma busca intensa (isto é, com uma vizinhança maior) na vizinhança de uma solução interessante. No caso presente, utilizou-se a a melhor solução conhecida como reinício e considerou-se a vizinhança para a busca intensiva:

- inserção de um elemento e remoção de dois;
- remoção de um elemento e inserção de dois;
- inserção de dois elementos e remoção de dois<sup>1</sup>;

## 3 Resultados Computacionais

A implementação foi feita com base na *framework* disponibilizada com a implementação base da *Tabu Search* na linguagem de programação java. O problema foi executado num ideapad S145 81S90005BR: Lenovo IdeaPad S145 Notebook Intel Core i5-8265U (6MB Cache, 1.6GHz), 8GB DDR4-SDRAM, 460 GB SSD, Intel UHD Graphics 620. O sistema operacional foi o Fedora 35 executando o Java 17 e Gradle 7. O código desenvolvido pode ser encontrado em [3].

As tabelas 1, 2, 3, 4, 5, 6 e 7 tem os resultados obtidos nos experimentos, variando as variações conforme solicitado nos requisitos desta atividade. A primeira coluna enumera cada instancia, a segunda coluna tem a identificação da instância atual, a terceira coluna a definição da busca local utilizada, a quarta coluna a variação do método e por fim na quinta e sexta coluna o tempo de execução e a melhor solução encontrada. Vale ressaltar que na variação do método,

---

<sup>1</sup>essa opção foi desabilitada nas instâncias de tamanho 400 pois consumiam muito tempo computacional.

	Instances	Local Search	Tenure Ratio	Method Variation	Execution Time	Best cost
0	020	best	0.2	default	1.263000	93*
1	020	best	0.2	intensification	0.447000	93*
2	020	best	0.2	diversification	0.246000	93*
3	020	best	0.4	default	0.997000	120*
4	020	best	0.4	intensification	0.476000	120*
5	020	best	0.4	diversification	0.255000	120*
6	020	first	0.2	default	0.043000	93*
7	020	first	0.2	intensification	0.035000	104*
8	020	first	0.2	diversification	0.031000	93*
9	020	first	0.4	default	0.057000	102*
10	020	first	0.4	intensification	0.205000	110*
11	020	first	0.4	diversification	0.035000	104*

Tabela 1: Solução obtida para cada configuração para a instancia kqbf020. O asterisco (\*) na coluna *Best Cost* indica que a solução está dentro do range de soluções ótimas.

*default* significa que não foi utilizada nenhuma estratégia de intensificação e diversificação, *intensification* significa que foi utilizada apenas *intensification by restart*, e *diversification* significa que foram utilizadas *intensification and diversification by restart*.

## 4 Análise

De acordo com os números apresentados, o cenário que mais se aproximou dos intervalos nos quais os valores das soluções ótimas eram esperados foi a variação *Intensification* com o *best-improving*.

É notável a diferença em tempo por iteração da estratégia *first-improving* em comparação com a *best-improving* para as maiores instâncias. O que é esperado devido o *first improving* fazer escolhas mais rápidas na média, em compensação as variações ficaram mais próximas do intervalo esperado com o *best-improving*.

A influência do parâmetro *Tenure* pode ser observada, onde as instâncias com 0.4 obtiveram melhor resultado do que instâncias com 0.2, na maioria dos casos, levando em consideração a busca local. Tal resultado é interessante pois mostra quanto o algoritmo deve ser restringida

	Instances	Local Search	Tenure Ratio	Method Variation	Execution Time	Best cost
12	040	best	0.2	default	2.660000	239
13	040	best	0.2	intensification	1.633000	290*
14	040	best	0.2	diversification	1.284000	260
15	040	best	0.4	default	2.707000	308*
16	040	best	0.4	intensification	4.774000	316*
17	040	best	0.4	diversification	1.218000	303*
18	040	first	0.2	default	0.049000	201
19	040	first	0.2	intensification	0.090000	239
20	040	first	0.2	diversification	0.195000	243
21	040	first	0.4	default	0.114000	239
22	040	first	0.4	intensification	0.147000	239
23	040	first	0.4	diversification	0.224000	239

Tabela 2: Solução obtida para cada configuração para a instancia kqbf040. O asterisco (\*) na coluna *Best Cost* indica que a solução está dentro do range de soluções ótimas.

	Instances	Local Search	Tenure Ratio	Method Variation	Execution Time	Best cost
24	060	best	0.2	default	5.390000	368
25	060	best	0.2	intensification	4.510000	480*
26	060	best	0.2	diversification	3.904000	483*
27	060	best	0.4	default	5.427000	491*
28	060	best	0.4	intensification	5.166000	446*
29	060	best	0.4	diversification	5.662000	491*
30	060	first	0.2	default	0.066000	413
31	060	first	0.2	intensification	0.238000	452*
32	060	first	0.2	diversification	0.715000	406
33	060	first	0.4	default	0.163000	408
34	060	first	0.4	intensification	0.327000	397
35	060	first	0.4	diversification	3.064000	455*

Tabela 3: Solução obtida para cada configuração para a instancia kqbf060. O asterisco (\*) na coluna *Best Cost* indica que a solução está dentro do range de soluções ótimas.

	Instances	Local Search	Tenure Ratio	Method Variation	Execution Time	Best cost
36	080	best	0.2	default	10.182000	683
37	080	best	0.2	intensification	7.984000	662
38	080	best	0.2	diversification	7.651000	780*
39	080	best	0.4	default	10.318000	783*
40	080	best	0.4	intensification	8.769000	732*
41	080	best	0.4	diversification	6.590000	702
42	080	first	0.2	default	0.089000	667
43	080	first	0.2	intensification	0.513000	674
44	080	first	0.2	diversification	2.313000	680
45	080	first	0.4	default	0.273000	692
46	080	first	0.4	intensification	0.649000	692
47	080	first	0.4	diversification	2.205000	692

Tabela 4: Solução obtida para cada configuração para a instancia kqbf080. O asterisco (\*) na coluna *Best Cost* indica que a solução está dentro do range de soluções ótimas.

	Instances	Local Search	Tenure Ratio	Method Variation	Execution Time	Best cost
48	100	best	0.2	default	17.558000	1220*
49	100	best	0.2	intensification	15.455000	1116*
50	100	best	0.2	diversification	130.585000	1193*
51	100	best	0.4	default	17.549000	1227*
52	100	best	0.4	intensification	28.078000	1249*
53	100	best	0.4	diversification	17.294000	1225*
54	100	first	0.2	default	0.118000	979*
55	100	first	0.2	intensification	1.041000	966*
56	100	first	0.2	diversification	5.663000	850
57	100	first	0.4	default	0.350000	1121*
58	100	first	0.4	intensification	1.519000	1128*
59	100	first	0.4	diversification	6.309000	878

Tabela 5: Solução obtida para cada configuração para a instancia kqbf100. O asterisco (\*) na coluna *Best Cost* indica que a solução está dentro do range de soluções ótimas.

	Instances	Local Search	Tenure Ratio	Method Variation	Execution Time	Best cost
60	200	best	0.2	default	123.439000	3947*
61	200	best	0.2	intensification	132.799000	3710*
62	200	best	0.2	diversification	202.997000	3529
63	200	best	0.4	default	160.816000	3710*
64	200	best	0.4	intensification	692.925000	3703*
65	200	best	0.4	diversification	444.594000	3782*
66	200	first	0.2	default	0.378000	3234
67	200	first	0.2	intensification	15.587000	3497
68	200	first	0.2	diversification	82.407000	3517
69	200	first	0.4	default	1.315000	3183
70	200	first	0.4	intensification	15.661000	3372
71	200	first	0.4	diversification	79.993000	3234

Tabela 6: Solução obtida para cada configuração para a instancia kqbf200. O asterisco (\*) na coluna *Best Cost* indica que a solução está dentro do range de soluções ótimas.

	Instances	Local Search	Tenure Ratio	Method Variation	Execution Time	Best cost
72	400	best	0.2	default	1007.581000	10547
73	400	best	0.2	intensification	927.970000	10816
74	400	best	0.2	diversification	974.393000	10760
75	400	best	0.4	default	1060.901000	10760
76	400	best	0.4	intensification	965.190000	10645
77	400	best	0.4	diversification	1116.701000	10758
78	400	first	0.2	default	1.556000	9391
79	400	first	0.2	intensification	3.751000	9550
80	400	first	0.2	diversification	14.326000	9222
81	400	first	0.4	default	3.369000	9237
82	400	first	0.4	intensification	6.118000	9211
83	400	first	0.4	diversification	14.747000	10254

Tabela 7: Solução obtida para cada configuração para a instancia kqbf400. O asterisco (\*) na coluna *Best Cost* indica que a solução está dentro do range de soluções ótimas.



no problema em mãos para que o processo de busca saia de mínimos locais e explore bem o espaço de solução.

## Referências

- [1] Michel Gendreau and Jean-Yves Potvin. Tabu search. In M. Gendreau and J.-Y. Potvin, editors, *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science*, chapter 2, pages 41–56. Springer Science+Business Media, 2010.
- [2] Gary Kochenberger, Jin-Kao Hao, Fred Glover, Mark Lewis, Zhipeng Lü, Haibo Wang, and Yang Wang. The unconstrained binary quadratic programming problem: A survey. *J. Comb. Optim.*, 28(1):58–81, jul 2014.
- [3] Ítalo Fernandes Gonçalves, Lucas Guesser Targino da Silva, and Luiz Gustavo Silva Aguiar. Implementação do max-qbf com ts. [https://github.com/lucasguesserts/MO824A-combinatorial-optimization/tree/activity-5/activity\\_5](https://github.com/lucasguesserts/MO824A-combinatorial-optimization/tree/activity-5/activity_5). Accessed: 2022-05-27.