

Metaheurísticas para Problemas de Otimização Combinatória: Dependência de Pacotes e Mochila com Penalidades

Fellipe Sanha

5 de dezembro de 2025

Resumo

Este trabalho apresenta a implementação e análise de metaheurísticas para dois problemas de otimização combinatória: o Problema da Dependência de Pacotes e o Problema da Mochila com Penalidades (KPF). São exploradas abordagens de Programação Linear Inteira (ILP) e o algoritmo BRKGA (Biased Random-Key Genetic Algorithm), com comparação de desempenho em instâncias de diferentes tamanhos.

Sumário

1	Introdução	2
2	Problema Abordado	2
2.1	Definição Formal	2
3	Soluções Exploradas	2
3.1	Programação Linear Inteira (ILP)	2
3.2	BRKGA (Biased Random-Key Genetic Algorithm)	3
3.2.1	Parâmetros Utilizados	3
4	Comparação de resultados	3
4.1	Instâncias de Teste	3
4.2	Comparação ILP vs BRKGA	4
5	Conclusão	4

1 Introdução

Neste trabalho abordaremos a descrição de variações de um NP difíceis, similar ao Problema da Mochila(Knapsack Problem - KP), estratégias viáveis para resolver este problema em diferentes recortes temporais, e comparamos os resultados obtidos frente aos resultados ótimos conhecidos na literatura.

2 Problema Abordado

O Problema da Mochila com Penalidades (Knapsack Problem with Forfeits - KPF) é uma variante do problema clássico da mochila onde pares de itens podem ter penalidades associadas quando ambos são selecionados (CERULLI et al., 2020). (MOURA; NORONHA; BOGUE, 2021) propuseram uma heurística ILS para este problema.

2.1 Definição Formal

Dado um conjunto de n itens, onde cada item i possui:

- Peso w_i
- Lucro p_i

E um conjunto de pares de forfeit F , onde cada par $f = \{i, j\} \in F$ possui uma penalidade d_f aplicada quando ambos os itens i e j são selecionados.

O objetivo é maximizar:

$$\max \sum_{i=1}^n p_i x_i - \sum_{f \in F} d_f v_f \quad (1)$$

Sujeito a:

$$\sum_{i=1}^n w_i x_i \leq H \quad (2)$$

$$x_i + x_j - v_f \leq 1, \quad \forall f = \{i, j\} \in F \quad (3)$$

$$x_i \in \{0, 1\}, \quad \forall i \quad (4)$$

$$v_f \in [0, 1], \quad \forall f \in F \quad (5)$$

3 Soluções Exploradas

3.1 Programação Linear Inteira (ILP)

A formulação ILP segue o modelo de Cerulli et al. (CERULLI et al., 2020), onde:

- $x_i = 1$ se o item i é selecionado
- v_f representa a ativação da penalidade do par de forfeit f

A implementação utiliza o solver HiGHS através da biblioteca JuMP em Julia, com warm-start baseado em soluções GRASP.

3.2 BRKGA (Biased Random-Key Genetic Algorithm)

O BRKGA (Biased Random-Key Genetic Algorithm) foi implementado com uma estratégia de codificação inteligente baseada em GRASP:

- **Codificação:** Chaves aleatórias no intervalo $[0, 1]$
- **Decodificação:** Estratégia de threshold onde itens com chave \geq threshold são candidatos à seleção
- **Warm-start:** Soluções iniciais geradas por heurística gulosa com diferentes valores de α

3.2.1 Parâmetros Utilizados

Parâmetro	Valor
Tamanho da população	1000
Threshold	0.5
α (GRASP)	0.7
Iterações máximas	50000
Tempo limite	60s

Tabela 1: Parâmetros do BRKGA

4 Comparação de resultados

Nesta seção veremos os desempenhos das diferentes estratégias desenvolvidas. Compararemos, também, os resultados com os ótimos conhecidos da literatura (MOURA; NORONHA; BOGUE, 2021).

4.1 Instâncias de Teste

Foram escolhidas vinte instâncias no total, de quatro tamanhos diferentes, 500 itens, 700 itens, 800 itens, e 1000 itens , visando avaliar o desempenho dos algoritmos desenvolvidos desde instâncias relativamente pequenas(500 itens) até tamanhos considerados grandes(1000 itens).

4.2 Comparação ILP vs BRKGA

Tamanho	Instância	Ótimo	60s		120s		180s	
			ILP	BRKGA	ILP	BRKGA	ILP	BRKGA
500	1	2626	2145	2244	2336	2244	2336	2244
	2	2660	2308	2238	2324	2238	2419	2242
	3	2516	2207	2152	2299	2163	2310	2198
	4	2556	2233	2170	2254	2170	2445	2185
	5	2625	2297	2199	2353	2199	2409	2218
700	1	3589	3156	3059	3218	3059	3299	3059
	2	3679	2763	2857	2775	2857	2913	2857
	3	3664	3128	3124	3189	3124	3206	3124
	4	3647	3182	3085	3267	3085	3267	3085
	5	3596	3188	3082	3156	3082	3128	3082
800	1	4184	3400	3398	3400	3427	3520	3427
	2	4065	3300	3371	3299	3371	3299	3371
	3	4104	3475	3316	3475	3388	3475	3388
	4	4056	3218	3349	3252	3349	3320	3349
	5	4086	3366	3389	3366	3389	3386	3389
1000	1	4940	3852	4028	3916	4077	3928	4077
	2	4969	4034	4105	4034	4105	4034	4105
	3	5177	4040	4230	4211	4230	4273	4230
	4	5143	4091	4235	4137	4302	4212	4302
	5	5136	4094	4143	4019	4145	4040	4145

Tabela 2: Resultados comparativos por tamanho de instância e tempo limite

Tamanho	60s		120s		180s	
	ILP	BRKGA	ILP	BRKGA	ILP	BRKGA
500	13.79	15.24	10.88	15.15	8.17	14.58
700	15.13	16.31	14.1	16.31	12.97	16.31
800	18.23	17.91	18.07	17.42	17.06	17.42
1000	19.81	18.23	20.22	17.76	19.82	17.76

Tabela 3: GAP comparativo por tamanho de instância

5 Conclusão

Referências

CERULLI, Raffaele et al. The Knapsack Problem with Forfeits. In: BAÏOU, Mourad et al. (Ed.). **Combinatorial Optimization**. Cham: Springer International Publishing, 2020. P. 263–272. DOI: 10.1007/978-3-030-53262-8_22.

MOURA, Ana Flávia Ciríaco; NORONHA, Thiago; BOGUE, Eduardo Theodoro. Uma heurística ILS para o Problema da Mochila com Penalidades. In: ANAIS do LIII Simpósio Brasileiro de Pesquisa Operacional. João Pessoa: Galoá, 2021. Disponível em: <<https://proceedings.science/sbpo/sbpo-2021/trabalhos/uma-heuristica-ils-para-o-problema-da-mochila-com-penalidades?lang=pt-br>>. Acesso em: 4 dez. 2025.