

***Random-Key Optimizer* para Problemas Inteiros Mistos: Estudo de Caso em Otimização de Portfólios**

Antonio A. Chaves¹, Mauricio G. C. Resende

Universidade Federal de São Paulo

antonio.chaves@unifesp.br, mgcr@berkeley.edu

RESUMO

Programação Inteira Mista (MIPs) abrange problemas de otimização NP-difíceis com variáveis inteiras e contínuas, aplicáveis em diversas indústrias, como logística, telecomunicações e finanças. Solucionadores comerciais (CPLEX, Gurobi, Xpress) usam principalmente o método *branch-and-bound*, mas são caros e podem ser ineficientes para problemas grandes. Este trabalho propõe o *Random-Key Optimizer* (RKO), um *framework* flexível onde o usuário desenvolve uma função de decodificação para lidar com restrições do MIP. Como prova de conceito, o método é aplicado ao problema de otimização de portfólio de Markowitz com restrições de compra e cardinalidade.

PALAVRAS CHAVE. Metaheurística. Programação Inteira Mista. Portifólio.

MH - Metaheurísticas

1. Introdução

Programação Inteira Mista (MIPs) contemplam uma ampla classe de problemas de otimização NP-difíceis onde o objetivo é minimizar alguma função objetivo sujeita a restrições, com algumas ou todas as variáveis restringidas a serem de valor inteiro [Karp, 2009; Papadimitriou and Steiglitz, 1998]. Aplicações de MIPs podem ser encontradas em diversas indústrias, em áreas como transporte e logística, telecomunicações, manufatura e finanças. Exemplos incluem planejamento de capacidade, alocação de recursos, problema da mochila e otimização de portfólio, entre outros [Korte et al., 2011]. Solucionadores comerciais como CPLEX, Gurobi ou Xpress são tipicamente usados para resolver MIPs, entretanto os usuários precisam pagar taxas de licenciamento caras e possuem conhecimentos limitados sobre os algoritmos utilizados. Além disso, esses solucionadores tipicamente dependem da estratégia *branch-and-bound* (B&B) que pode resolver problemas de pequeno e médio porte eficientemente até a otimalidade, mas frequentemente possuem tempos de execução exponenciais no tamanho da entrada, exigindo o desenvolvimento de métodos novos e eficientes que possam lidar com a escala cada vez maior de problemas de otimização do mundo real.

Neste trabalho, mostramos como o *Random-Key Optimizer* (RKO) [Chaves et al., 2024] pode ser usado para resolver MIPs. Especificamente, definimos um *framework* com diversas heurísticas e metaheurísticas gerais, no qual o usuário é responsável por desenvolver apenas uma função de decodificação que trate as restrições do MIP, como restrições de integralidade, limites de variáveis e restrições de cardinalidade, e mapeia uma solução de *random-key* em uma solução do problema abordado. Para fins de prova de conceito, neste artigo testamos nossa abordagem com experimentos numéricos para o problema de otimização de portfólio de média-variância de Markowitz, na presença de restrições de compra e cardinalidade.

¹Este trabalho é financiado pela FAPESP (2024/08848-3) e CNPq (305557/2024-6)

2. Random-Key Optimizer (RKO)

Nossa abordagem utiliza o conceito de chaves aleatórias (*random-keys*), proposto no Biased Random-Key Genetic Algorithm (BRKGA) [Gonçalves and Resende, 2011]. Para estabelecer nossa notação e terminologia, iniciamos com uma breve revisão do BRKGA e mostramos como esse algoritmo pode ser generalizado para estratégias de soluções alternativas, como o *Simulated Annealing*, entre outras. Um refinamento do RKGa proposto por Bean [Bean, 1994], o BRKGA apresenta uma estrutura heurística para a resolução de problemas de otimização. Embora a maior parte do trabalho na literatura tenha se concentrado em problemas de otimização combinatória, o BRKGA também tem sido aplicado a problemas de otimização contínua [Silva et al., 2014]. O BRKGA baseia-se na ideia de que qualquer solução para um problema de otimização pode ser codificada como um vetor de chaves aleatórias, ou seja, um vetor $\mathcal{X}$ no qual cada entrada é um número real, gerado aleatoriamente no intervalo $(0, 1)$. Tal vetor $\mathcal{X}$, definido no hipercubo $[0, 1]^n$, é mapeado para uma solução viável para o problema de otimização original com a ajuda do *decoder*, um algoritmo determinístico que recebe como entrada um vetor de chaves aleatórias e retorna uma solução viável para o problema de otimização, bem como o custo da solução.

O BRKGA foi generalizado para um paradigma geral de *Random-Key Optimizer* (RKO) [Chaves et al., 2024]. Assim como o BRKGA, o RKO usa um vetor de chaves aleatórias para codificar uma solução para um problema de otimização combinatória e utiliza um *decoder* para avaliar uma solução codificada pelo vetor de chaves aleatórias. No entanto, no RKO, a busca independente do problema no espaço de chaves aleatórias é generalizada para além do algoritmo genético, em direção a uma classe muito maior de metaheurísticas, usando (por exemplo) simulated annealing (SA), greedy randomized adaptive search procedure (GRASP), iterated local search (ILS), variable neighbourhood search (VNS), particle swarm optimization (PSO), genetic algorithm (GA), and large neighbourhood search (LNS).

3. Problema de otimização de portfólio

Neste artigo consideramos o problema canônico de Markowitz (média-variância) para seleção de portfólios, incluindo restrições de cardinalidade que limitam o portfólio a ter um número específico de ativos e limites nas proporções de investimento em cada ativo (quando o ativo é incluído) [Chang et al., 2000; Cesarone et al., 2011]. Essas restrições visam limitar a exposição ao risco e os custos de transação para pequenos investimentos, por meio de limites superiores e inferiores, respectivamente. Assim, mostramos que nossa abordagem também é capaz de lidar com MIPs não lineares dentro do mesmo *framework*.

Seja n o número total de ativos disponíveis, μ_i o retorno esperado do ativo $i = 1, \dots, n$, Σ_{ij} a covariância entre os ativos $i = 1, \dots, n$ e $j = 1, \dots, n$, e λ o parâmetro de aversão ao risco, com $0 \leq \lambda \leq 1$. Além disso, seja K o número desejado de ativos no portfólio, l_i a proporção mínima que deve ser mantida do ativo $i = 1, \dots, n$ (se algum investimento for realizado no ativo), e u_i a proporção máxima permitida para o ativo $i = 1, \dots, n$, com $0 \leq l_i \leq u_i \leq 1$. Na prática, os limites inferiores l_i representam um nível mínimo de compra ou transação para o ativo i , enquanto os limites superiores u_i limitam a exposição do portfólio a ativos individuais. As variáveis de decisão são os pesos de alocação w_i para o ativo $i = 1, \dots, n$, com $0 \leq w_i \leq 1$, juntamente com as variáveis indicadoras binárias $z_i \in \{0, 1\}$, onde $z_i = 1$ se o ativo i for incluído no portfólio, e $z_i = 0$ caso contrário. Em seguida, consideramos o seguinte problema de otimização inteira mista:

$$\min_{\mathbf{w}} H(\mathbf{w}) = \lambda \mathbf{w}^\top \Sigma \mathbf{w} - (1 - \lambda) \boldsymbol{\mu}^\top \mathbf{w} \quad (1)$$

$$\text{s.a.} \quad \sum_{i=1}^n w_i = 1, \quad (2)$$

$$\sum_{i=1}^n z_i = K, \quad (3)$$

$$l_i z_i \leq w_i \leq u_i z_i, \quad i = 1, \dots, n, \quad (4)$$

$$z_i \in \{0, 1\}, \quad i = 1, \dots, n. \quad (5)$$

No caso limite $\lambda = 0$, o objetivo é maximizar o retorno esperado (independentemente do risco), resultando em uma solução ótima que geralmente inclui um único ativo com maior retorno. Por outro lado, para $\lambda = 1$, o foco é minimizar o risco (independentemente do retorno), o que geralmente leva a uma solução com um número maior de ativos. Valores intermediários entre 0 e 1 descrevem um compromisso explícito entre risco e retorno.

O problema apresenta quatro famílias de restrições: a restrição 2 garante que os recursos disponíveis sejam totalmente alocados; a restrição 3 força a seleção de um portfólio com exatamente K ativos. A restrição 4 garante que todos os investimentos não nulos permaneçam dentro da faixa de investimento definida pelos limites inferiores l e superiores u , conectando as variáveis inteiras z_i às variáveis contínuas w_i . Por fim, a restrição 5 impõe que as variáveis z_i sejam binárias, atuando como variáveis indicadoras.

O Algoritmo 1 define um *decoder* para o problema de otimização de portfólio de Markowitz considerando as restrições do modelo.

Algoritmo 1: *Decoder* para problema de otimização de portfólio

Entrada: $\mu, \Sigma, \lambda, l, u, K, \mathcal{X}$

Saída: $w, z, custo$

para $i \leftarrow 1$ **até** K **faça**

$z[i] \leftarrow \text{int}(\mathcal{X}[i] \times n)$ // Obtém os ativos selecionados (restrição 3 e 5)

$w[i] \leftarrow l[i] + (u[i] - l[i]) \times \mathcal{X}[i]$ // Define os pesos de alocação (restrição 4)

para $i \leftarrow 1$ **até** K **faça**

$w[i] \leftarrow w[i] / \sum_i w_i$ // Normaliza os pesos (restrição 2)

$penalidade \leftarrow \max(0, w[i] - u_i) + \max(0, l_i - w[i])$

$custo \leftarrow \lambda \mathbf{w}^\top \Sigma \mathbf{w} - (1 - \lambda) \boldsymbol{\mu}^\top \mathbf{w} + 10^7 \times penalidade$ // Custo da solução

retorna $w, z, custo$

A seguir, mostramos a viabilidade e o desempenho do RKO por meio de experimentos com dados disponíveis na OR-Library. A Tabela 1 contém informações sobre as instâncias (n , K , l_i , u_i e λ), os resultados do Gurobi (limitante superior UB, limitante inferior LB, gap e tempo de execução - limitado a 1800 segundos) e do RKO (melhor solução, desvio relativo RPD e tempo para encontrar a melhor solução). Utilizando o decoder descrito no Alg. 1, verificamos que o RKO é capaz de resolver de forma eficiente o MIP não linear abordado, atendendo a todas as restrições (orçamentária, de cardinalidade, de limites e de integralidade) e alcançando qualidade de solução compatível com a solução exata gerada pelo Gurobi. É importante destacar que o RKO gera soluções em tempo polinomial, ao contrário dos tempos exponenciais geralmente observados para solucionadores exatos baseados em *branch and bound*, o que o torna uma opção interessante para problemas de otimização em grande escala. No entanto, observamos que, embora em alguns casos as soluções do RKO possam coincidir com a solução exata do problema, em geral, a abordagem produz soluções aproximadas que podem ser suficientes em termos de qualidade para aplicações práticas.

Tabela 1: Resultados computacionais para o problema de otimização de portfólio.

Instância	n	K	l_i	u_i	λ	Gurobi				RKO			
						UB	LB	gap(%)	tempo (s)	OFV	RPD _b	RPD _a	tempo (s)
port1 (Hang Seng)	31	5	0,01	0,25	0,3	-0,00466	-0,00466	0,00	0,02	-0,00466	0,00	0,01	6,64
port2 (DAX)	85	10	0,01	0,25	0,3	-0,00532	-0,00532	0,00	0,03	-0,00532	0,00	0,05	19,89
port3 (FTSE)	89	10	0,01	0,25	0,3	-0,00448	-0,00448	0,00	0,03	-0,00448	0,02	0,02	19,47
port4 (S&P)	98	10	0,01	0,25	0,3	-0,00549	-0,00549	0,00	0,04	-0,00549	0,00	0,01	19,78
port5 (Nikkei)	225	10	0,01	0,25	0,3	-0,00230	-0,00230	0,00	0,12	-0,00230	0,02	0,02	28,45
port6 (S&P 500)	457	20	0,01	0,25	0,3	-0,00888	-0,00888	0,00	338,80	-0,00888	0,03	0,04	49,94
port7 (Russell 2000)	1318	30	0,01	0,25	0,3	-0,01226	-0,01239	1,05	1808,74	-0,01224	0,16	0,38	99,87
port8 (Russell 3000)	2151	40	0,01	0,25	0,3	-0,00725	-0,00732	0,86	1814,92	-0,00709	2,27	2,31	199,67

Neste trabalho propomos um método baseado em representação de soluções por *random-keys*, *Random-Key Optimizer* (RKO), para resolver problemas de programação inteira mista (MIPs) em grande escala, aplicáveis a diversas áreas, como otimização de portfólio, planejamento de produção e roteirização de veículos. Os usuários geralmente dependem de solucionadores comerciais de caixa-preta, que envolvem altos custos de licenciamento. O RKO é um algoritmo de código aberto disponível no *Github*².

Referências

- Bean, J. C. (1994). Genetic algorithms and random keys for sequencing and optimization. *ORSA journal on computing*, 6(2):154–160.
- Cesarone, F., Scozzari, A., and Tardella, F. (2011). Portfolio selection problems in practice: a comparison between linear and quadratic optimization models. *arXiv preprint arXiv:1105.3594*.
- Chang, T.-J., Meade, N., Beasley, J. E., and Sharaiha, Y. M. (2000). Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research*, 27(13): 1271–1302.
- Chaves, A. A., Resende, M. G., Schuetz, M. J., Brubaker, J. K., Katzgraber, H. G., de Aruda, E. F., and Silva, R. (2024). A random-key optimizer for combinatorial optimization. *arXiv preprint arXiv:2411.04293*.
- Gonçalves, J. F. and Resende, M. G. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17(5):487–525.
- Karp, R. M. (2009). Reducibility among combinatorial problems. In *50 Years of Integer Programming 1958-2008: from the Early Years to the State-of-the-Art*, p. 219–241. Springer.
- Korte, B. H., Vygen, J., Korte, B., and Vygen, J. (2011). *Combinatorial optimization*, volume 1. Springer.
- Papadimitriou, C. H. and Steiglitz, K. (1998). *Combinatorial optimization: algorithms and complexity*. Courier Corporation.
- Silva, R. M., Resende, M. G., and Pardalos, P. M. (2014). Finding multiple roots of a box-constrained system of nonlinear equations with a biased random-key genetic algorithm. *Journal of Global Optimization*, 60:289–306.

²<https://github.com/RKO-solver>