São José dos Campos - SP
LV SBPO
LV Simpósio Brasileiro de Pesquisa Operacional
São José dos Campos, SP - 6 a 9 de novembro de 2023

Progressive Hedging aplicado a um problema de otimização de portfólio de dois estágios

Eugênio Silva Rezende, Elisangela Martins de Sá, Sérgio Ricardo de Souza
Centro Federal de Educação Tecnológica de Minas Gerais - CEFET-MG
Av. Amazonas, 7.675, Bairro Nova Gameleira, Belo Horizonte/MG
eugeniomatcomp@gmail.com, elisangelamartins@cefetmg.br,
sergio@cefetmg.br

RESUMO

A otimização de portfólio é um problema importante na área de finanças, que busca determinar um portfólio que otimize um determinado objetivo. Este trabalho propõe um algoritmo baseado no método Progressive Hedging para resolver um modelo de otimização de portfólio baseado em programação estocástica. Experimentos computacionais foram realizados para verificar o desempenho e qualidade das soluções obtidas pelo algoritmo proposto. Neste trabalho, é mostrado que o algoritmo proposto foi capaz de determinar a solução ótima para um conjunto de instâncias. Além disso, os resultados obtidos mostraram que o algoritmo proposto gasta muito menos tempo computacional do que o solver para resolver instâncias com muitos cenários.

PALAVRAS CHAVE. Otimização de portfólio, *Progressive Hedging*, Programação Estocástica.

Tópicos: EST&MP – Estatística e Modelos Probabilísticos e OC – Otimização Combinatória

ABSTRACT

Portfolio optimization is an important problem in the area of finance, which seeks to design a portfolio to optimize a given objective. This work proposes an algorithm based on the Progressive Hedging method to solve a portfolio optimization model based on stochastic programming. Computational experiments were carried out to verify the efficiency and quality of the solutions of the proposed method. In this work, it is shown that the proposed method was able to determine the optimal solution for a set of instances. Furthermore, the obtained results showed that the proposed method spends much less computational time than the solver to solve the instances with many scenarios.

KEYWORDS. Portfolio Optimization. Progressive Hedging. Stochastic Programming.

Paper topics (indicate in order of PRIORITY the paper topic(s))

DOI: 10.59254/sbpo-2023-174880
http://live-proceedings.pantheonsite.io/p/174880

1. Introdução

O problema de otimização de portfólio consiste em selecionar um conjunto de ativos para comprar ou vender de forma a maximizar ou minimizar um determinado objetivo. Os principais objetivos são maximizar o retorno do portfólio, restringindo-se o risco, minimizar o risco do portfólio, impondo um nível mínimo de retorno, ou otimizar um objetivo que incorpora a maximização de retornos e a minimização de riscos.

O trabalho de Markowitz [MarNowitz, 1952] foi o primeiro a propor um modelo de otimização de portfólio. Nesse modelo, a medida de risco usada é a variância. Embora o modelo de Markowitz seja ainda importante, ele apresenta algumas características indesejáveis. Entre essas características do modelo está a ausência de restrições que representem os custos de transações de compra e venda. Essas restrições são importantes, pois representam situações reais do mercado, em que compras e vendas de ativos envolvem custos adicionais. Outra importante característica indesejável do modelo de Markowitz está na medida de risco utilizada: a variância, como medida de risco, captura tanto oscilações positivas como negativas do retorno. Idealmente, seria importante apenas considerar as oscilações negativas do retorno, pois estas representam situações em que há perda no investimento. Por fim, outro fator que representa uma desvantagem do modelo de Markowitz é que os retornos médios dos ativos e as covariâncias entre esses são parâmetros fixos e conhecidos, desconsiderando fatores de incerteza.

Depois do modelo de Markowitz, diversos trabalhos apresentaram modificações nesse modelo, como Konno e Wijayanayake [2001], Metaxiotis e Liagkouras [2012] e Crama e Schyns [2003]). Essas modificações tinham o objetivo de criar modelos mais realistas que o modelo de Markowitz. Como um exemplo desses modelos modificados, pode-se considerar o modelo apresentado por Cui et al. [2020]. Nesse modelo, o problema de otimização de portfólio é formulado como um problema de programação estocástica, que inclui restrições de custo de transação, utilização do CVaR como medida de risco e utilização de cenários de preços dos ativos para modelar as incertezas do mercado.

Para resolver problemas de otimização de portfólio, vários métodos computacionais foram desenvolvidos. Para o modelo de Markowitz, que consiste em um problema de programação quadrática, há métodos exatos disponíveis. Para o modelo de Cui et al. [2020], foi desenvolvido um algoritmo baseado em uma metaheurística populacional de Aprendizagem Incremental. Os mesmos autores, em um trabalho anterior, desenvolveram também um método baseado em um algoritmo genético para resolver o mesmo modelo [Cui et al., 2015].

Este trabalho tem, como objetivo, apresentar um algoritmo para resolver o problema de otimização de portfólio, que produza boas soluções em um tempo computacional aceitável para problemas que consideram muitos cenários de incertezas. Neste trabalho, utilizou-se um modelo biobjetivo, obtido por uma modificação do modelo de Cui et al. [2020]. Nesse modelo, o risco, medido pelo CVaR, e o retorno médio aparecem na função objetivo. Para resolver o modelo utilizado, empregou-se o algoritmo *Progressive Hedging* (PH). Esse algoritmo foi proposto originalmente por Rockafellar e Wets [1991]. Neste trabalho, modificou-se o algoritmo original e aplicou-se o PH modificado ao modelo de otimização de portfólio.

Este artigo está dividido da seguinte forma: na seção 2, é apresentado o problema de otimização de portfólio e sua formulação. Na seção 3, é apresentada uma formulação do subproblema do PH e também são abordados os detalhes sobre as modificações feitas no algoritmo PH original. Na seção 4 são apresentados os detalhes do experimentos computacionais e os resultados. Por fim, na seção 5 é feita uma conclusão deste trabalho.

DOI: 10.59254/sbpo-2023-174880
http://live-proceedings.pantheonsite.io/p/174880

2. Descrição do Problema e Formulação

Esta seção apresenta uma descrição do problema e a notação e a formulação matemática utilizadas.

2.1. Descrição do Problema

O problema de otimização abordado neste trabalho consiste em determinar a alocação de recursos em um portfólio de N ativos distintos, de modo que o risco relacionado ao investimento seja o mínimo possível e que o retorno seja o máximo possível. Nesse problema, há momentos em que o gestor do portfólio pode modificar a sua composição, por meio da compra e venda de ativos. Porém, as ações do gestor devem ser sempre balizadas de acordo com o objetivo de minimizar o risco e maximizar o retorno.

No problema estudado neste trabalho, considera-se que alocações dos recursos do portfólio ocorrem somente em dois momentos distintos - também referidos como estágios. No primeiro estágio de alocação dos recursos, sabe-se os preços dos ativos e dispõem-se de um quantia h de recursos para a compra daqueles. O segundo estágio de alocação dos recursos corresponde a um tempo futuro, em que os preços dos ativos aparecem como realizações de variáveis aleatórias. Além disso, é imposto ao problema uma limitação na quantidade de ativos que podem fazer parte do portfólio em qualquer momento. Com isso, considera-se que no primeiro e segundo estágios somente K ativos possam fazer parte do porfólio. Além dessas restrições, exige-se também uma quantidade mínima para cada ativo que compõe o portfólio e uma quantidade mínima que pode ser comprada ou vendida.

O desafio do problema aqui abordado é que as decisões de alocação do primeiro e do segundo estágio devem ser tomadas no primeiro estágio, considerando-se a natureza aleatória dos futuros preços dos ativos, que somente são conhecidos no segundo estágio. Uma das formas de modelar a natureza aleatória dos preços dos ativos no segundo estágio é considerar vários cenários possíveis de preço, com uma determinada probabilidade de ocorrência para cada um. Na otimização, essa abordagem de modelar os eventos aleatórios como cenários com probabilidade atribuída a cada um é característica da programação estocástica. Na próxima seção, será apresentado um modelo de programação estocástica inteiro-misto para o problema apresentado nesta seção, assim como mais detalhes da modelagem desse problema.

2.2. Notação e Formulação

A notação da formulação para o problema abordado neste trabalho é a seguinte. Considera-se T estágios. Neste trabalho considera-se dois estágios, então, $T = 2$. No entanto, o modelo apresentado pode ser facilmente estendido para vários estágios.

A formulação matemática utiliza $i \in \{1, \ldots, N\}$ como índice do ativo, em que N é o número de ativos; $t \in \{1, ..., T\}$ como índice do estágio, em que T é o número de estágios; e $s \in \{1, \ldots, S\}$ como índice do cenário, em que S é o número de cenários. Os parâmetros usados são h_t, que representa os recursos financeiros para compra de ativos em t; K, que representa a quantidade de ativos distintos permitida em qualquer estágio; $w_{\min}$, que representa a quantidade mínima de qualquer ativo que pode compor o portfólio; $t_{\min}$, que representa a quantidade mínima de qualquer ativo que pode ser negociada; P_{its}, que representa o preço do ativo i, no estágio t e cenário s; $\eta_{b'}$, que representa o custo fixo de compra dos ativos; $\eta_{s'}$, que representa o custo fixo de venda dos ativos; $\rho_{b'}$, que representa o custo por quantidade de ativo comprado; $\rho_{s'}$, que representa o custo por quantidade de ativo vendido; μ, que representa o retorno desejado do portfólio; M, que é uma constante numérica de valor significativo; P_{its}, que representa o preço do ativo i no estágio t e cenário s; e, por fim, β, que é o valor associado ao risco utilizado no cálculo do CVaR.

DOI: 10.59254/sbpo-2023-174880
http://live-proceedings.pantheonsite.io/p/174880

As variáveis de decisão do modelo são as seguintes: w_{its} representa a quantidade do ativo i em posse, no estágio t e cenário s; b_{its} representa a quantidade do ativo i comprado, no estágio t e cenário s; s_{its} representa a quantidade do ativo i vendido, no estágio t e cenário s; c_{its} é uma variável binária, de valor igual a $c_{its} = 1$ se o ativo i entra na composição do portfólio no estágio t e cenário s, e $c_{its} = 0$, caso contrário; f_{its} é uma variável binária igual de valor $f_{its} = 1$ se o ativo i é comprado no estágio t e cenário s, e $f_{its} = 0$, caso contrário; g_{its} é uma variável binária de valor igual a $g_{its} = 1$ se o ativo i é vendido no estágio t e cenário s, e $g_{its} = 0$, caso contrário; $\hat{\alpha}$ e z_s são variáveis de decisão contínuas originadas pela linearização do CVaR.

A formulação para o problema é então dada por:

$$\min \quad \hat{\alpha} + \frac{1}{(1-\beta)} \sum_{k=1}^{S} p_s z_s - \theta \sum_{k=1}^{S} p_s R_s \tag{1}$$

$$\text{s. a: } w_{i(t+1)s} = w_{its} + b_{its} - s_{its}, \quad i = 1, ..., N, t = 1, ..., T, s = 1, ..., S \tag{2}$$

$$h_t + \sum_{i=1}^{N} s_{its} P_{its} - (\eta_{s'} g_{its} + s_i \rho_{s'} P_{its}) = \tag{3}$$

$$\sum_{i=1}^{N} b_{its} P_{its} - (\eta_{b'} f_{its} + b_i \rho_{b'} P_{its}) \quad t = 1, ..., T, s = 1, ..., S$$

$$\sum_{i=1}^{N} c_{its} = K, \quad t = 1, ..., T, s = 1, ..., S \tag{4}$$

$$w_{min} c_{its} \leq w_{its}, \quad i = 1, ..., N, t = 1, ..., T, s = 1, ..., S \tag{5}$$

$$t_{min} f_{its} \leq b_{its}, \quad i = 1, ..., N, t = 1, ..., T, s = 1, ..., S \tag{6}$$

$$t_{min} g_{its} \leq s_{its}, \quad i = 1, ..., N, t = 1, ..., T, s = 1, ..., S \tag{7}$$

$$f_{its} + g_{its} \leq 1, \quad i = 1, ..., N, t = 1, ..., T, s = 1, ..., S \tag{8}$$

$$w_{its} \leq M c_{its}, \quad i = 1, ..., N, t = 1, ..., T, s = 1, ..., S \tag{9}$$

$$f_{its} M \leq b_{its}, \quad i = 1, ..., N, t = 1, ..., T, s = 1, ..., S \tag{10}$$

$$g_{its} M \leq s_{its}, \quad i = 1, ..., N, t = 1, ..., T, s = 1, ..., S \tag{11}$$

$$b_{its} M \leq f_{its}, \quad i = 1, ..., N, t = 1, ..., T, s = 1, ..., S \tag{12}$$

$$s_{its} M \leq g_{its}, \quad i = 1, ..., N, t = 1, ..., T, s = 1, ..., S \tag{13}$$

$$R_s = \sum_{i=1}^{N} P_{iTs} w_{iTs} - \sum_{i=1}^{N} P_{i0s} w_{i0s}, \quad s = 1, ..., S \tag{14}$$

$$z_s \geq -R_s - \alpha_s, s = 1, ..., S \tag{15}$$

$$z_s \geq 0, \quad s = 1, ..., S \tag{16}$$

$$w_{its} = w_{it(s+1)}, b_{its} = b_{it(s+1)}, \quad i = 1, ..., N, t = 1, ..., T-1, s = 1, ..., S-1 \tag{17}$$

$$s_{its} = s_{it(s+1)}, f_{its} = f_{it(s+1)}, \quad i = 1, ..., N, t = 1, ..., T-1, s = 1, ..., S-1 \tag{18}$$

$$g_{its} = g_{it(s+1)}, c_{its} = c_{it(s+1)}, \quad i = 1, ..., N, t = 1, ..., T-1, s = 1, ..., S-1 \tag{19}$$

$$\alpha_s = \alpha_{s+1}, \quad s = 1, ..., S-1 \tag{20}$$

$$w_{its}, b_{its}, s_{its} \in \mathbb{R}^+ \tag{21}$$

$$c_{its}, f_{its}, g_{its} \in \{0, 1\}. \tag{22}$$

DOI: 10.59254/sbpo-2023-174880
http://live-proceedings.pantheonsite.io/p/174880

Neste modelo, a função objetivo (1) busca minimizar o CVaR e maximizar o retorno do porfólio. As restrições (2) garantem o balanceamento da quantidade de ativos em cada período. As restrições (3) garantem o balanceamento de fluxo financeiro em cada período. As restrições (4) representam a imposição de que, em qualquer estágio do problema, somente K ativos possam compor o portfólio. As restrições (5)–(7) impõem limites mínimos de posse, de compra e de venda dos ativos. As restrições (8) representam o impedimento de que um tipo específico de ativo seja comprado e vendido em um mesmo estágio. Já as restrições (9)–(13) acomplam as variáveis binárias às variáveis contínuas. O conjunto de restrições (14)–(16) está relacionado ao cálculo do CVaR. As restrições (14) representam o cálculo do retorno do portfólio para o cenário s e as restrições (15)-(16) são oriundas da linearização do cálculo do CVaR. As restrições (17)–(20) representam as decisões não antecipativas do modelo. Por fim, os conjuntos de restrições (21) e (22) indicam os domínios das variáveis do modelo.

3. *Progressive Hedging* aplicado ao Problema

Nesta seção, apresenta-se a formulação do subproblema do *Progressive Hedging* (PH). Em seguida, apresenta-se as modificações feitas no algoritmo original.

3.1. Formulação do subproblema

O *Progressive Hedging* (PH) é um algoritmo, proposto por Rockafellar e Wets [1991], frequentemente usado na resolução de problemas de programação estocástica. Esse algoritmo resolve problemas de programação estocástica por meio de uma decomposição do problema original. O PH, na sua versão original, decompõe o problema original em vários subproblemas, cada um desses associado a um único cenário. Essa decomposição decorre da relaxação lagrangeana das restrições de não antecipatividade, que acoplam variáveis de decisão pertencentes a cenários diferentes. Estas restrições garantem que as decisões não podem se basear na antecipação de valores que os parâmetros incertos possam assumir no futuro. Desta forma, para o problema com dois estágios, as decisões de primeiro estágio devem ser as mesmas para todos os cenários. Ao relaxar estas restrições, são adicionadas à função objetivo penalizações pelo não atendimento delas. Este novo componente penaliza o afastamento do valor das variáveis de primeiro estágio, associadas às restrições não antecipatórias, do valor médio desta variável na última iteração para o conjunto de cenários. Desta forma, espera-se que estas variáveis vão convergir para os mesmos valores ao final da execução do algoritmo.

Neste trabalho, são definidos subproblemas para grupos com S' cenários. Assim, será denotada por $\mathbf{X}'$ a região factível determinada por esses cenários, retirando-se as restrições de não antecipatividade, que são as restrições indicadas pelas expressões (17)–(20). O modelo para cada subproblema utiliza os vetores de multiplicadores lagrangeanos $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_S)$ e $\boldsymbol{\lambda^j} = (\lambda^j_{1,1}, \ldots, \lambda^j_{N,S})$, $j \in \{2, 3, 4, 5, 6, 7\}$, associados as restrições de não antecipatividade relativas às variáveis α_s, w_{i1s}, c_{i1s}, b_{i1s}, s_{i1s}, f_{i1s} e g_{i1s}. O modelo incorpora as variáveis de decisão α_s^{k+1}, w_{its}^{k+1}, c_{its}^{k+1}, b_{its}^{k+1}, s_{its}^{k+1}, f_{its}^{k+1} e g_{its}^{k+1} na $(k+1)$-ésima iteração do algoritmo. Os parâmetros $\hat{\alpha}^k, \hat{w}_i^k, \hat{c}_i^k, \hat{b}_i^k, \hat{s}_i^k, \hat{f}_i^k$ e $\hat{g}_i^k$ são as soluções médias, calculadas na iteração k, sobre todos os cenários, ponderados pelas probabilidades destes. Como exemplo, calcula-se $\hat{w}_i^k$ por:

$$\hat{w}_i^k = \sum_{s=1}^{S} p_s w_{i1s}^k, \tag{23}$$

em que w_{i1s}^k é o valor da variável de decisão calculada na k-ésima iteração. Os parâmetros $\rho_{j,k}$, para $j \in \{1, 2, 3, 4, 5, 6, 7\}$, são os parâmetros de penalidade, calculados na k-ésima iteração, relacionados à restrição de não antecipatividade de cada uma das variáveis do modelo.

DOI: 10.59254/sbpo-2023-174880
http://live-proceedings.pantheonsite.io/p/174880

Procedendo-se com a relaxação dessas restrições, obtém-se a seguinte formulação para cada subproblema na iteração $(k+1)$ do PH:

$$\begin{aligned}\min \quad & \hat{\alpha} + \frac{1}{(1-\beta)} \sum_{k=1}^{S'} p_s z_s - \theta \sum_{k=1}^{S'} p_s R_s + \sum_{s=1}^{S'} \lambda_s^{1,k} \left(\alpha_s^{k+1} - \hat{\alpha^k}\right) + \sum_{s=1}^{S'} \sum_{i=1}^{N} \lambda_{is}^{2,k} \left(w_{i0s}^{k+1} - \hat{w}_i^k\right) \\ & + \sum_{s=1}^{S'} \sum_{i=1}^{N} \lambda_{is}^{3,k} \left(c_{i0s}^{k+1} - \hat{c}_i^k\right) + \sum_{s=1}^{S'} \sum_{i=1}^{N} \lambda_{is}^{4,k} \left(b_{i0s}^{k+1} - \hat{b}_i^k\right) \sum_{s=1}^{S'} \sum_{i=1}^{N} \lambda_{is}^{5,k} \left(s_{i0s}^{k+1} - \hat{s}_i^k\right) \\ & + \sum_{s=1}^{S'} \sum_{i=1}^{N} \lambda_{is}^{6,k} \left(f_{i0s}^{k+1} - \hat{f}_i^k\right) + \sum_{s=1}^{S'} \sum_{i=1}^{N} \lambda_{is}^{7,k} \left(g_{i0s}^{k+1} - \hat{g}_i^k\right) + \rho_{1,k} \sum_{s=1}^{S'} ||\alpha_s^{k+1} - \hat{\alpha}^k||^2 \\ & + \rho_{2,k} \sum_{s=1}^{S'} \sum_{i=1}^{N} ||w_{i0s}^{k+1} - \hat{w}_i^k||^2 + \rho_{3,k} \sum_{s=1}^{S'} \sum_{i=1}^{N} ||c_{i0s}^{k+1} - \hat{c}_i^k||^2 + \rho_{4,k} \sum_{s=1}^{S'} \sum_{i=1}^{N} ||b_{i0s}^{k+1} - \hat{b}_i^k||^2 \\ & + \rho_{5,k} \sum_{s=1}^{S'} \sum_{i=1}^{N} ||s_{i0s}^{k+1} - \hat{s}_i^k||^2 + \rho_{6,k} \sum_{s=1}^{S'} \sum_{i=1}^{N} ||f_{i0s}^{k+1} - \hat{f}_i^k||^2 + \rho_{7,k} \sum_{s=1}^{S'} \sum_{i=1}^{N} ||g_{i0s}^{k+1} - \hat{g}_i^k||^2 \\ \text{sujeito a: } & (\alpha_s^{k+1}, w_{its}^{k+1}, c_{its}^{k+1}, b_{its}^{k+1}, s_{its}^{k+1}, f_{its}^{k+1}, g_{its}^{k+1}) \in \mathbf{X}'\end{aligned} \tag{24}$$

3.2. Adaptação do PH

Neste trabalho, foram feitas adaptações ao algoritmo PH original [Rockafellar e Wets, 1991]. Todas as adaptações feitas amparam-se em adaptações descritas em trabalhos da literatura nos quais o PH foi utilizado como método de resolução de outros problemas de programação estocástica. A seguir, lista-se essas adaptações, explicando o funcionamento de cada uma delas:

(i) **Fixação de variáveis binárias**: as variáveis c_{i1s} de todos os cenários são fixadas no valor 1, para um determinado ativo i, quando, em alguma iteração do algoritmo, temos:

$$\frac{\sum_{s=1}^{S} c_{i1s}}{S} = p, \tag{25}$$

em que $p \in (0, 1]$. Nos trabalhos que utilizam o PH, essa modificação é geralmente utilizada em problemas com variáveis binárias [Garcia-Gonzalo et al., 2020; Watson e Woodruff, 2011].

(ii) **Adaptação dos parâmetros de penalidade**: os parâmetros de penalidade das variáveis w_{i1s}, c_{i1s} e α_s são atualizados ao longo da execução do algoritmo. Para atualizar esses parâmetros, verifica-se, durante a execução do algoritmo, o estado de de outros parâmetros, que estão relacionados a aspectos de convergência do algoritmo. Para apresentar a forma de atualização dos parâmetros de penalidade, considere que ρ_{k+1} seja um parâmetro de penalidade genérico, a ser calculado na $(k+1)$-ésima iteração. Para atualizar ρ_{k+1}, faz-se:

$$\rho_{k+1} = \begin{cases} \delta \rho_k, & \text{para } \Delta_D^k - \Delta_D^{k-1} > 0, \Delta_P^k - \Delta_P^{k-1} < 0 \\ \frac{1}{\delta} \rho^k, & \text{para } \Delta_P^k - \Delta_P^{k-1} > 0, \Delta_D^k - \Delta_D^{k-1} < 0 \\ \rho^k, & \text{Caso contrário} \end{cases}$$

Para mais detalhes sobre os parâmetros Δ_D^k, Δ_D^{k-1}, Δ_P^k e Δ_P^{k-1}, recomenda-se a leitura de Gul [2010].

DOI: 10.59254/sbpo-2023-174880
http://live-proceedings.pantheonsite.io/p/174880

(iii) **Resolução do Problema Determinístico Equivalente**: o Problema Determinístico Equivalente (PDE) é o problema original, com todos os cenários e restrições de não antecipatividade. Neste trabalho, escolheu-se por resolver o PDE assim que a quantidade de variáveis c_{i1s} fixadas atingisse K ativos. Dizendo de outra forma, resolveu-se o PDE assim que:

$$\frac{\sum_{s=1}^{S} c_{i1s}}{S} = p, \tag{26}$$

para os ativos cujos índices são o conjunto A, tal que $A \subset \{1, \ldots, N\}$ e $|A| = K$. Um exemplo de trabalho que emprega esse tipo modificação pode ser encontrado em Peng et al. [2019].

(iv) **Critério de parada modificado**: seja $x^k = (x_1^k, \ldots, x_\Omega^k)$ o vetor de variáveis de decisão da k-ésima iteração do PH, em que o componente x_ω^k, $\omega = 1, \ldots, \Omega$ representa que a variável de decisão é ligada ao cenário ω. Seja também $\hat{x}$ a solução média de x, dada por:

$$\hat{x}^k = \sum_{\omega=1}^{\Omega} p_\omega x_\omega^k, \tag{27}$$

em que p_ω é a probabilidade de ocorrência do cenário ω. O algoritmo PH original para assim que:

$$\sqrt{\sum_{\omega \in \Omega} p_\omega \left\| x_\omega^{k+1} - \hat{x}_\omega^k \right\|_2^2} < \epsilon, \tag{28}$$

para um valor de ϵ escolhido *a priori*. Considerando o que foi exposto, designe-se o lado esquerdo da desigualdade (28) como a distância média d_{x^k} de x^k em relação à solução média da $(k-1)$-ésima iteração. Neste trabalho, o critério de parada adotado foi o seguinte: o algoritmo para assim que:

$$\frac{d_{w^k} + d_{s^k} + d_{b^k} + d_{c^k} + d_{f^k} + d_{g^k} + d_{\alpha^k}}{d_{w^1} + d_{s^1} + d_{b^1} + d_{c^1} + d_{f^1} + d_{g^1} + d_{\alpha^1}} < \epsilon. \tag{29}$$

Portanto, pela expressão (29), o algoritmo para sempre quando a razão entre a soma das distâncias médias das variáveis é menor que um ϵ escolhido *a priori*. Outra possibilidade de parada do algoritmo se dá quando o PDE é resolvido.

(v) **Inclusão dos termos de penalidade na primeira iteração**: no PH original, na primeira iteração resolve-se cada subproblema sem a adição dos termos de penalidade. Já no algoritmo implementado neste trabalho, os termos de penalidade, ligados às restrições de não antecipatividade, são adicionados na primeira iteração, devido à inviabilidade dos subproblemas.

4. Experimentos computacionais

Os experimentos computacionais foram divididos em duas etapas. Na primeira etapa, resolveu-se instâncias para definir os melhores parâmetros de penalidade para as variáveis w_{i1s} e α_s. Por sua vez, na segunda etapa, resolveu-se outro conjunto de instâncias com os parâmetros definidos na primeira etapa. Para a realização dos experimentos computacionais, nas duas etapas, resolveu-se três conjuntos distintos de instâncias. Na primeira parte dessa seção, são apresentados de forma detalhada os conjuntos de instâncias criados. Em seguida, é apresentada a forma como foram escolhidos os valores iniciais para os multiplicadores lagrangeanos. Posteriormente, são apresentados os resultados sobre a pesquisa dos parâmetros de penalidade, obtidos na primeira etapa dos experimentos computacionais. Depois, são apresentados os recursos computacionais utilizados neste trabalho. Por fim, serão apresentados os resultados dos experimentos computacionais finais.

DOI: 10.59254/sbpo-2023-174880
http://live-proceedings.pantheonsite.io/p/174880

Tabela 1: Parâmetros das instâncias resolvidas.

Grupo	N	S	K	h_0	M	T	μ	β	$\eta_{b'}$	$\eta_{s'}$	$\rho_{b'}$	$\rho_{s'}$	w_{min}	t_{min}
A	20	20	1	100	1.000	2	2	0,90	0,1	0,1	0,1	0,1	0,01	0,01
B	40	100	1	100	1.000	2	2	0,90	0,1	0,1	0,1	0,1	0,01	0,01
C	40	1.000	1	100	1.000	2	2	0,90	0,1	0,1	0,1	0,1	0,01	0,01

4.1. Instâncias

Para as duas etapas dos experimentos computacionais, foram criadas instâncias, que foram divididas em três grupos, denominados A, B e C, respectivamente. Além disso, para cada grupo de instâncias foram criadas 10 instâncias de teste, para a primeira etapa de calibração dos parâmetros de penalidade, e outras 30 instâncias, que foram resolvidas na segunda etapa dos experimentos computacionais. A Tabela 1 apresenta os parâmetros das instâncias utilizadas neste trabalho.

Os preços dos ativos no primeiro estágio foram fixados em 100 para todas as instâncias. Para o segundo estágio, foram gerados S preços aleatórios do conjunto $\{p \in \mathbb{N} | 90 \leq p \leq 119\}$ para cada ativo.

4.2. Parâmetros

4.2.1. Multiplicadores lagrangeanos iniciais

Neste trabalho, escolheu-se por iniciar os multiplicadores lagrangeanos da primeira iteração do algoritmo implementado com o valor 0.

4.2.2. Pesquisa dos valores dos parâmetros de penalidade

Ao se aplicar o PH para resolver diferentes problemas de programação estocástica, constata-se que, em geral, os resultados obtidos são sensíveis ao parâmetro de penalidade inicial escolhido [Zehtabian e Bastin, 2016]. Por essa razão, neste trabalho, dividiu-se os experimentos computacionais em duas etapas, sendo a primeira delas dedicada à pesquisa dos melhores parâmetros de penalidade iniciais. Entende-se nesse trabalho como melhores parâmetros de penalidade aqueles cujos resultados, quando aplicados na resolução das instâncias, mostraram menores valores de *gap*. Por sua vez, define-se o gap como

$$gap = 100 \times \frac{FO_{PH} - FO_{solver}}{\left|FO_{PH}\right|}\%, \tag{30}$$

em que FO_{PH} é o valor da função objetivo obtido da aplicação do PH e FO_{solver} é o valor da função objetivo quando aplica-se o *solver*.

Na primeira etapa, portanto, o estudo consistiu na procura pelos melhores valores de parâmetros de penalidade iniciais para os três grupos de instâncias criados. O primeiro passo nesse estudo foi resolver as 10 instâncias de cada um dos grupos utilizando-se o PH e o *solver*, variando-se os parâmetros $\rho_{1,0}$ e $\rho_{2,0}$ do PH - os parâmetros de penalidade das demais variáveis de decisão foram mantidos fixos com o valor 1. Cabe aqui pontuar que, tanto na primeira como na segunda etapa dos experimentos desse trabalho, foram dadas às instâncias um tempo máximo de 1 hora para suas resoluções. A Tabela 2 mostra os parâmetros de penalidade testados para os três conjuntos de instâncias.

Após variar os parâmetros de penalidade e resolver as instâncias de teste, foram obtidos resultados para cada um dos grupos. Os resultados dos grupos A, B e C são apresentados, respectivamente, nas Tabelas 3, 4 e 5. Nessas tabelas, a coluna $\rho_{2,0}$ se refere ao parâmetro de penalidade inicial relacionado às variáveis w_{i1s} e a coluna $\rho_{1,0}$ ao parâmetro de penalidade inicial das variáveis α_s. Já a coluna "$\#gap \leq 5\%$" se refere à quantidade de instâncias resolvidas cujos valores de *gap*

DOI: 10.59254/sbpo-2023-174880
http://live-proceedings.pantheonsite.io/p/174880

Tabela 2: Parâmetros de penalidade iniciais testados para os três conjuntos de instâncias

Grupo	$\rho_{1,0}$	$\rho_{2,0}$
A	$\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$	$\{1, 2, 5, 10\}$
B	$\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$	$\{10^{-3}, 10^{-2}, 10^{-1}, 1\}$
C	$\{10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$	$\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$

Tabela 3: Quantidade de instâncias do grupo A que foram resolvidas e cujos *gaps* obtidos foram menores que 5%

$\rho_{2,0}$	$\rho_{1,0}$	# *gap* $\leq 5\%$
1	0.0001	2
	0.0010	2
	0.0100	1
	0.1000	1
2	0.0001	2
	0.0010	3
	0.0100	1
	0.1000	2
5	0.0001	3
	0.0010	2
	0.0100	1
	0.1000	2
10	0.0001	1
	0.0010	1
	0.0100	1
	0.1000	2

mostraram valores menores que 5%. Cabe ressaltar que as combinações de parâmetros que não resultaram em nenhuma instância com *gap* não excedente a 5% foram omitidas das tabelas.

Ao se analisar os resultados da primeira etapa do grupo A, percebe-se que, para as duplas de valores $(\rho_{1,0}, \rho_2) = (10^{-3}, 2)$ e $(\rho_{1,0}, \rho_2) = (10^{-4}, 5)$, obteve-se duas instâncias resolvidas cujos valores de *gaps* não excederam 5%. Em relação ao grupo B, a utilização da dupla de parâmetros $(\rho_{1,0}, \rho_2) = (10^{-3}, 10^{-2})$ resultou em três instâncias resolvidas cujos valores de *gaps* não excederam 5%. Já em relação ao grupo C, observa-se, pela Tabela 5, variados casos em que até duas instâncias resolvidas tiveram *gap* que não excedeu 5%.

A partir da análise sobre os resultados obtidos da primeira etapa, resolveu-se utilizar os seguintes valores para os parâmetros de penalidade para cada um dos grupos:

- **Grupo A**: $(\rho_{1,0}, \rho_{2,0}) = (10^{-4}, 5)$;
- **Grupo B**: $(\rho_{1,0}, \rho_{2,0}) = (10^{-3}, 10^{-2})$;
- **Grupo C**: $(\rho_{1,0}, \rho_{2,0}) = (10^{-3}, 10^{-2})$.

4.3. Recursos computacionais

Neste trabalho, os experimentos foram executados em computador com processador Intel Core i7-10510U com frequência 1.80GHz, 16 GB de memória RAM disponíveis e sistema operacional Ubuntu, versão 20.04, instalado. A versão utilizada do CPLEX foi a 12.9.0.

DOI: 10.59254/sbpo-2023-174880
http://live-proceedings.pantheonsite.io/p/174880

Tabela 4: Quantidade de instâncias do grupo B que foram resolvidas e cujos *gap* obtidos foram menores que 5%

$\rho_{2,0}$	$\rho_{1,0}$	$\#gap \leq 5\%$
1.000000e-4	1.000000e-6	3
	1.000000e-5	3
	1.000000e-4	2
	1.000000e-3	1
1.000000e-3	1.000000e-4	1
1.000000e-2	1.000000e-6	1
	1.000000e-4	1
	1.000000e-3	4
1.000000e-1	1.000000e-6	2
	1.000000e-5	2
1.0000	1.000000e-06	1
	1.000000e-04	1

Tabela 5: Quantidade de instâncias do grupo C que foram resolvidas e cujos *gaps* obtidos foram menores que 5%

$\rho_{2,0}$	$\rho_{1,0}$	# *gap* $\leq 5\%$
0.0001	1.000000e-07	1
	1.000000e-06	1
0.0010	1.000000e-07	1
	1.000000e-06	1
	1.000000e-05	2
	1.000000e-04	1
	1.000000e-03	2
0.0100	1.000000e-06	1
	1.000000e-05	1
	1.000000e-04	1
	1.000000e-03	2
0.1000	1.000000e-07	2
	1.000000e-06	1
	1.000000e-05	1
1.0000	1.000000e-06	1
	1.000000e-05	1

4.4. Resultados e Discussão

Na segunda etapa, foram resolvidas 30 instâncias de cada um dos grupos criados. Os parâmetros de penalidade iniciais utilizados foram aqueles apresentados ao final da seção anterior. Com esses parâmetros, foi aplicado o PH modificado, implementado neste trabalho, às instâncias. Ademais, também aplicou-se o *solver* CPLEX para resolver as instâncias. Os resultados obtidos nessa etapa estão contidos na Tabela 6. Nessas tabelas, as colunas "Tempo médio (s) PH"e "Tempo médio *solver*"contêm, respectivamente, os tempos de resolução obtidos na aplicação do PH modificado e do *solver*. Por fim, a coluna "*gap* mediano"contem os gap mediano obtido com o PH modificado, utilizando-se a fórmula (30).

Tabela 6: Tempos médios de resolução obtidos pelo *solver*, pelo PH modificado e *gap* mediano obtido desse algoritmo

Grupo	*gap* mendiano (%)	Tempo médio (s) PH	Tempo médio (s) *solver*
A	98,07	7,63	0,00
B	36,62	34,36	139,73
C	12,83	249,8	3494,66

Analisando-se os resultados do grupo A da Tabela 6, observa-se que o *gap* mediano obtido é alto. Porém, em 10 instâncias das 30, obteve-se um gap inferior a 5%. Em todos esses 10 casos,

DOI: 10.59254/sbpo-2023-174880
http://live-proceedings.pantheonsite.io/p/174880

o *gap* obtido foi de $0,00\%$. Observa-se também que, para esse grupo, o tempo de resolução do PH modificado foi consideravelmente inferior ao tempo de resolução do *solver*.

Partindo-se para a análise do grupo B, percebe-se que *gap* mediano é significativamente inferior ao obtido para o grupo A. Para esse grupo, somente 5 instâncias resolvidas tiveram valores de *gap* que não excederam 5%. Além disso, nota-se que somente em 4 instâncias o *gap* foi de $0,00\%$. Outra constatação importante feita a partir dessa tabela é que, para todas as instâncias desse grupo, o tempo de resolução do PH modificado é consideravelmente inferior ao tempo de resolução do *solver*. Para se ter uma ideia dessa discrepância, o tempo médio do PH modificado representa cerca de 25% do tempo médio do *solver*.

Com relação aos resultados do grupo C, observa-se que o *gap* mediano desse grupo é significativamente inferior ao obtido para o grupo B. Além disso, contabilizou-se a quantidade de instâncias resolvidas cujos valores de *gap* não excederam o valor de 5%, como foi feito para os outros grupos de instâncias. Nesse quesito, foram 6 instâncias. Ademais, somente contabilizou-se um único caso em que o *gap* foi de 0.00%. Como surpresa, para instância 4-1000-1, observou-se um valor de *gap* negativo, indicando que a solução utilizando-se o PH modificado foi menor que a solução obtida pelo *solver*. Partindo-se para analise do tempo de resolução, observa-se por essa tabela uma eficiência maior do PH modificado em relação *solver* que a encontrada no grupo B de instâncias. Fazendo-se os cálculos com os dados da tabela, verifica-se que o tempo médio de resolução do PH modificado representa cerca de 7% do tempo de resolução do *solver*.

5. Conclusões

Neste trabalho, implementou-se uma versão modificada do PH para resolver um modelo de otimização de portfólio biobjetivo. Para avaliar o algoritmo implementado, foram criadas instâncias e realizados experimentos computacionais.

A primeira parte dos experimentos computacionais foi dedicada à pesquisa dos parâmetros de penalidade iniciais. Para isso, foram resolvidas 10 instâncias de teste de cada grupo de instâncias. Ao final dessa etapa, estabeleceu-se os melhores parâmetros para a segunda etapa de experimentos computacionais.

Na segunda etapa dos experimentos computacionais, resolveu-se 30 instâncias de cada grupo. Ao analisar os resultados dessa etapa, observou-se que o *gap* mediano obtido foi mais alto para o grupo A de instâncias. Observou-se também que esse *gap* reduziu-se à medida que foram resolvidas instâncias com mais cenários. Outra importante observação é a de que o tempo médio de execução do PH modificado reduziu significativamente à medida que se resolveram instâncias com mais cenários.

Como trabalho futuro, sugere-se a implementação de modificações que tornem o PH menos sensível ao parâmetro inicial de penalidade, com o intuito de melhorar a qualidade das soluções encontradas, sem aumentar o tempo de resolução. Outra possibilidade de trabalho é explorar combinações do PH com outros métodos resolução.

Referências

Crama, Y. e Schyns, M. (2003). Simulated annealing for complex portfolio selection problems. *European Journal of operational research*, 150(3):546–571.

Cui, T., Bai, R., Ding, S., Parkes, A. J., Qu, R., He, F., e Li, J. (2020). A hybrid combinatorial approach to a two-stage stochastic portfolio optimization model with uncertain asset prices. *Soft Computing*, 24:2809–2831.

Cui, T., Bai, R., Parkes, A. J., He, F., Qu, R., e Li, J. (2015). A hybrid genetic algorithm for a two-stage stochastic portfolio optimization with uncertain asset prices. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, p. 2518–2525. IEEE.

DOI: 10.59254/sbpo-2023-174880
http://live-proceedings.pantheonsite.io/p/174880

Garcia-Gonzalo, J., Pais, C., Bachmatiuk, J., Barreiro, S., e Weintraub, A. (2020). A progressive hedging approach to solve harvest scheduling problem under climate change. *Forests*, 11(2):224.

Gul, S. (2010). Optimization of surgery delivery systems. Technical report, Arizona State University.

Konno, H. e Wijayanayake, A. (2001). Portfolio optimization problem under concave transaction costs and minimal transaction unit constraints. *Mathematical Programming*, 89:233–250.

MarNowitz, H. (1952). Portfolio selection. *Journal of Finance*, 7(1):77–91.

Metaxiotis, K. e Liagkouras, K. (2012). Multiobjective evolutionary algorithms for portfolio management: A comprehensive literature review. *Expert Systems with Applications*, 39(14):11685–11698.

Peng, Z., Zhang, Y., Feng, Y., Rong, G., e Su, H. (2019). A progressive hedging-based solution approach for integrated planning and scheduling problems under demand uncertainty. *Industrial & Engineering Chemistry Research*, 58(32):14880–14896.

Rockafellar, R. T. e Wets, R. J.-B. (1991). Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of operations research*, 16(1):119–147.

Watson, J.-P. e Woodruff, D. L. (2011). Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. *Computational Management Science*, 8(4):355.

Zehtabian, S. e Bastin, F. (2016). Penalty parameter update strategies in progressive hedging algorithm. Tech report CIRRELT-2016-12, CIRRELT, Université de Montréal, Canadá.

DOI: 10.59254/sbpo-2023-174880
http://live-proceedings.pantheonsite.io/p/174880