Universidade Federal de Pernambuco

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

CENTRO DE INFORMÁTICA

2012.1

Uso do algoritmo de Colônia Artificial de Abelhas para problemas de otimização do portfólio

Trabalho de Graduação

Aluno : Victor Lorena de Farias Souza {vlfs@cin.ufpe.br}
Orientador: Adriano Lorena Inácio de Oliveira {alio@cin.ufpe.br}

25 de Junho de 2012

Universidade Federal de Pernambuco

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

CENTRO DE INFORMÁTICA

2012.1

Uso do algoritmo de Colônia Artificial de Abelhas para problemas de otimização do portfólio

Trabalho de Graduação

Trabalho de graduação apresentado no Centro de Informática da Universidade Federal de Pernambuco por Victor Lorena de Farias Souza, orientado por Adriano Lorena Inácio Oliveira, como requisito para a obtenção do Grau de Bacharel em Ciências da Computação.

Orientador: Adriano Lorena Inácio de Oliveira {alio@cin.ufpe.br}

25 de Junho de 2012

Assinaturas

Adriano Lorena Inácio de Oliveira
Orientador

Victor Lorena de Farias Souza
Aluno

"Ninguém vai bater mais forte do que a vida.
Não importa como você vai bater e sim o quanto aguenta apanhar e continuar lutando; o quanto pode suportar e seguir em frente. É assim que se ganha".

Anderson Silva

Agradecimento:

Aos meus pais e irmã que são pessoas que acreditam em mim e no meu potencial. Eles me deram força para me fazer seguir em frente, mesmo nos momentos mais complicados deste curso.

Aos professores e monitores desta universidade que sempre se dedicaram a formar profissionais diferenciais para o mercado de trabalho e, sem dúvida, fazem isso com muita competência.

Aos meus amigos de turma que me fizeram companhia durante estes quatro anos e meio de faculdade com quem compartilhei minhas viradas de noite, debates sobre diversos temas, contribuindo bastante para minha formação acadêmica.

A minha namorada, pela paciência e compreensão que teve que ter pelas várias vezes que não pude estar ao seu lado.

A todos, o meu MUITO OBRIGADO.

Resumo:

Existem vários estudos sobre o problema de otimização de portfólio e sobre o uso de técnicas heurísticas na investigação do portfólio ideal. Porém, quase nenhum desses estudos trabalha com o algoritmo de colônia artificial de abelhas (CAA). No Modelo de restrição da cardinalidade da variância média (CCMV), a otimização de portfólio é um problema de programação quadrática mista e inteiro (não linear) para o qual algoritmos eficientes não existem. O objetivo deste trabalho foi aplicar a técnica de CAA para o problema de otimização de portfólio. O conjunto de dados de testes utilizado foi formado pelos preços semanais dos índices: Hang Seng, de Hong Kong, DAX 100 da Alemanha, FTSE 100 do Reino Unido, S&P 100 dos EUA e Nikkei 225 do Japão, entre março de 1992 e setembro de 1997. Os resultados desse estudo mostraram que o CAA obteve melhores resultados que os demais algoritmos de otimização comparados: algoritmo genético, Busca tabu, Simulated Annealing, PSO. Pois, para as ações Hang Seng, FTSE 100 e S&P 100 tanto o valor da média de distância euclidiana quanto os valores dos erros de variância e de retorno foram menores, o que demonstra que os valores obtidos são melhores. Já nas ações DAX 100 e Nikkei 225, mesmo o CAA tendo obtido menores valores para a distância, obteve maiores valores para os erros de variância e de retorno, com isso não se pode dizer que os resultados foram melhores que os dos demais algoritmos, mas que foi tão favorável quanto eles. A partir dos resultados obtidos conclui-se que o algoritmo de colônia artificial de abelhas é bem sucedido em otimização de portfólio.

Palavras-chave: Colônia artificial de abelhas, Otimização de portfólio, Modelo de restrição da cardinalidade da variância média, Modelo Padrão de Markowitz.

Abstract:

There are many studies on the problem of portfolio optimization and the use of heuristic techniques for investigating the optimal portfolio. However, almost none of these studies work with the algorithm of artificial bee colony (CAA). In the cardinality constrained mean–variance model (CCMV), the portfolio optimization problem is a mixed-integer quadratic programming (nonlinear) for which efficient algorithms do not exist. The objective of this study was to apply the technique of CAA to the problem of portfolio optimization. The set of test data used is formed by the weekly price indices: Hang Seng, Hong Kong, DAX 100 in Germany, FTSE 100 UK, S & P 100 U.S. and Nikkei 225 in Japan between March 1992 and September 1997. The results of this study showed that CAA had better results than other optimization algorithms compared: genetic algorithm, tabu search, simulated annealing, PSO. Because for the stock Hang Seng, FTSE 100 and S&P 100 both the value of the Mean Euclidian distance and the values of the variance error and return error were lower, showing that the values are better. However, for the actions DAX 100 and Nikkei 225, even the CAA obtained lower values for the distance, showed higher values for the variance error and return error, with it, cannot be said that the results were better than the other algorithms but that was as favorable as they are. From the results it is concluded that the algorithm of artificial bee colony is successful in portfolio optimization.

Keywords: Artificial bee colony, Portfolio optimization, Cardinality constrained mean–variance, Markowitz standard model

Índice

ASSINATURAS **3**
1. INTRODUÇÃO **13**
1.1 contexto e motivação: 13
1.2 Objetivos: 14
1.3 Estrutura: 15
1.4 Trabalhos relacionados: 15
1.4.1. Algoritmos genéticos: 15
1.4.2. Busca Tabu 17
1.4.3. Simulated Annealing 18
1.4.4. Otimização por enxame de partículas 18
2. FORMULAÇÃO DO PROBLEMA **20**
2.1. Problema da seleção de portfólio 20
2.2. Modelo de Markowitz 20
2.2.1. Problema sem restrição 21
2.2.2. Fronteira eficiente 22
2.3. Modelo CCMV 24
2.3.1. Modelo com restrição: 24
3. SOLUÇÃO PROPOSTA **26**
3.1. Algoritmo de Colônia Artificial de Abelhas: 26
3.2. CAA para solução do CCMV 28
3.2.1. Função Fitness (aptidão) 30
3.2.2. Produção inicial das fontes de comida 31
3.2.3. Mandando abelha empregada para fonte de alimento 31
3.2.4. Abelhas seguidoras 32
3.2.5. Critério de abandono 33
4. EXPERIMENTOS COMPUTACIONAIS **35**
5. CONCLUSÃO **39**
6. REFERÊNCIAS **40**

Lista de símbolos.

CCMV Modelo de restrição da cardinalidade da variância média

AG Algoritmo genético (genetic algorithm)

TS Busca Tabu (tabu search)

SA Arrefecimento Simulado (simulated annealing)

PSO Otimização por enxame de partículas (particle swarm optimization)

CAA Colônia Artificial de Abelhas (Artificial Bee Colony)

FE Fronteira Eficiente

Lista de imagens:

Figura 1: Fronteira eficiente relativa a ação FTSE 100 [4]

Figura 2: Representação do algoritmo CAA [28]

Lista de algoritmos:

Algoritmo 1: Pseudocódigo para o algoritmo AG

Algoritmo 2: Pseudocódigo para o algoritmo TS

Algoritmo 3: Pseudocódigo para o algoritmo SA

Algoritmo 4: Pseudocódigo para o algoritmo PSO

Algoritmo 5: Pseudocódigo para o algoritmo CAA

Algoritmo 6: Organizar abelha

Algoritmo 7: CAA para problema de portfólio

Lista de tabelas:

Tabela 1: Resultados experimentais das heurísticas

1. Introdução

1.1 Contexto e motivação:

No mercado financeiro, portfólio, ou carteira, é o conjunto de ativos, títulos ou ações que os investidores possuem. Nesse cenário, surge o problema de otimização de portfólio que consiste na busca de uma diversificação nos investimentos do portfólio que ofereça o maior retorno para cada nível de risco ou o menor risco para o lucro desejado, ou seja, este problema é evidenciado pelos investidores ao alocar seu dinheiro entre as diferentes ações presentes no mercado. [1]

A metodologia básica para a otimização de portfólio começou com o clássico trabalho de Markowitz *"Portfolio Selection*" em 1952 [2]. Sua teoria propôs um problema de programação quadrática em que a função de risco, medido pela variância (conhecida como variância média) do retorno do portfólio, é minimizado de acordo com a restrição sobre o retorno médio ou o retorno é maximizado de acordo com a variância do retorno. Mesmo sendo considerada como a base do portfólio financeiro moderno, este modelo não tem sido extensivamente utilizado na prática. Em primeiro lugar, este quadro variância média é conhecido por ser válida se as taxas de retorno forem variáveis aleatórias que sigam a distribuição normal ou se a função do investidor utilidade é quadrática. Em segundo lugar, a evidência empírica indica que é muito difícil de resolver um problema de média variância de portfólio em larga escala com software existente para programação quadrática [3].

Mais recentemente, surgiram novos modelos que visam aprovar a ferramenta de medida de risco. Konno e Yamazaki (1991) [3] propôs o desvio absoluto, enquanto Speranza (1993) [3] propôs o desvio semi-absoluto como medidas de risco em vez de variância. King e Jensen (1992) [3], Markowitz (1991) [3] e Markowitz (1993) [3], adotam semi-variância como medida de risco. Hamza e Janssen (1995) [3] introduziram uma nova função de risco medida por uma combinação convexa das duas semivariâncias: da taxa de portfólio e do retorno a partir do valor médio. Hamza e Janssen (1996) [3] apresentaram uma abordagem linear para resolver problemas de grande porte do portfólio de otimização usando funções de risco assimétricas.

Atualmente, nota-se que o modelo mais utilizado é o Modelo de restrição da cardinalidade da variância média – CCMV (2008) [4,5]. Neste, é considerado que as

restrições de cardinalidade impõem um limite sobre o número de ativos no portfólio e as restrições de quantidade restringem a proporção de cada ativo no portfólio.

Devido às restrições de cardinalidade, os algoritmos tradicionais não são aconselhados a serem usados. Porém, existem alguns métodos heurísticos de resolução do "Modelo de Restrição da Cardinalidade da Variância Média". Estes métodos consistem no uso de algoritmos de otimização como por exemplo: algoritmos genéticos [4, 6], Busca Tabu [4], Simulated Annealing [4], a otimização por enxame de partículas [5, 7].

Nesse estudo é proposto a utilização de outro método heurístico, conhecido como Colônia Artificial de Abelhas, para a resolução do problema de otimização do portfólio. O CAA, proposto por Karaboga [7], é um algoritmo de inteligência coletiva baseada no comportamento de uma colônia de abelhas. A partir de pesquisas anteriores, mostrou-se que o algoritmo CAA supera outros métodos heurísticos em diversas áreas [8, 9, 10, 11, 12]. Além disso, ainda existem poucos estudos sobre o uso desse algoritmo e menos ainda do uso deste na área de otimização de portfólios.

1.2 Objetivos:

A abordagem padrão da média variância para seleção de portfólios envolve traçar uma fronteira eficiente, ou seja, uma curva contínua que ilustra a relação entre retorno e risco (variância). Esta fronteira pode ser facilmente encontrada através de programação quadrática. Esta abordagem é bem conhecida e largamente aplicada. No entanto, para fins práticos, pode ser desejável limitar o número de ações num portfólio, bem como a imposição de limites na proporção de qualquer ativo em particular dentro do portfólio. Se essas restrições existem, encontrar a fronteira eficiente se torna muito mais difícil. Este documento ilustra como é a aplicação heurística do CAA a esse problema, na presença de tais limitações. Além de mostrar resultados computacionais apresentados para um número de conjuntos de dados disponíveis.

1.3 Estrutura:

Esta monografia está organizada da seguinte forma: inicialmente, serão mostrados o contexto e a motivação para a realização do estudo, além dos objetivos que se desejam alcançar. Logo após, serão apresentados de forma resumida os modelos e definições envolvidos nos algoritmos, já presentes na literatura, que servirão de base para o estudo comparativo. Em seguida, serão apresentados os conceitos da formulação do problema, onde serão mostrados os modelos matemáticos de Markowitz e do CCMV. A seção 3, conterá as definições necessárias para o entendimento do algoritmo CAA, além da apresentação do modelo utilizado. Ao final, serão mostrados os resultados obtidos no estudo. Além de algumas conclusões e possíveis trabalhos futuros.

1.4 Trabalhos relacionados:

Foi observado na literatura que já existem alguns métodos heurísticos de resolução do CCMV. Estes métodos consistem no uso de algoritmos de otimização como, por exemplo: algoritmo genético, Busca Tabu, Simulated Annealing e PSO.

Estudos mostram que o Algoritmo de Colônia Artificial de abelhas apresentam melhores resultados se comparados aos algoritmos listados acima [8, 9, 10, 11, 12]. Além de não haver quase nenhum trabalho na literatura apresentando uma abordagem heurística para o problema de otimização de portfólio utilizando a técnica CAA. A partir dessa informação, observa-se a lacuna para a utilização do algoritmo das abelhas no problema de achar o portfólio ideal.

Nessa seção serão descritos, de forma resumida, esses algoritmos, cujos resultados serão comparados com os resultados obtidos com esse estudo.

1.4.1. Algoritmos genéticos:

O algoritmo genético (AG) pode ser descrito como um “Algoritmo de busca probabilística inteligente”. Esse modelo foi proposto originalmente por Holland [4, 6]. AG baseiam-se no processo evolutivo de organismos biológicos na natureza. Durante a evolução, as populações naturais evoluem de acordo com os princípios da “Seleção

Natural" e da "Sobrevivência do mais apto". Os indivíduos mais bem sucedidos na adaptação ao seu ambiente terão melhor chance de sobreviver e se reproduzir, enquanto indivíduos que são menos aptos serão eliminados. Isto significa que os genes dos indivíduos altamente adaptados vão se espalhar para um número crescente de indivíduos em cada geração sucessiva. A combinação de características boas dos pais altamente adaptados podem produzir descendentes ainda mais aptos. Deste modo, as espécies tendem a tornar-se cada vez melhor adaptados ao seu ambiente.

O AG simula esses processos, tendo uma população inicial de indivíduos e aplicando operadores genéticos em cada reprodução. Em termos de otimização, cada indivíduo da população é codificado em uma string ou cromossomo que representa uma possível solução para um dado problema. A aptidão de um indivíduo é avaliada em relação a uma função de determinado objetivo (chamada função fitness). Aos indivíduos ou soluções com mais aptidão são dadas oportunidades para se reproduzir através da troca de sua informação genética, em um procedimento chamado crossover, com outros indivíduos de fitness elevado. Isso produz novas soluções, que compartilham algumas características tomadas de ambos os pais. Associado a esse processo também existe a operação de mutação. Essa operação é frequentemente aplicada após o cruzamento, alterando alguns genes. As novas soluções podem substituir toda a população (abordagem geracional) ou substituir apenas os indivíduos menos aptos (abordagem estado estacionário). Este ciclo de avaliação-seleção-reprodução é repetido até que uma solução satisfatória seja encontrada. Os passos básicos de um AG são descritos da seguinte forma [6] : (Algoritmo 1).

Algoritmo 1: Pseudocódigo para o algoritmo AG
Gerar uma população inicial Avaliar a aptidão dos indivíduos na população Enquanto: Selecione os pais da população Reunir pares (pais) para produzirem crianças Avaliar a aptidão dos filhos Substituir alguns ou a totalidade da população pelas crianças Até que uma solução satisfatória foi encontrada.

1.4.2. Busca Tabu

A Busca Tabu (TS) é uma heurística de busca local, proposta por Glover e Hansen [13]. No TS o conceito fundamental é o de um "movimento": um operador sistemático que, dada uma solução única inicial gere uma série de outras possíveis soluções. Em termos de pesquisa locais, essas outras soluções são a "vizinhança" da solução única partida. Estas soluções podem, ou não, ser viáveis. A partir da vizinhança da melhor solução é escolhida uma nova solução para se tornar a partida para a iteração seguinte e os processos repetem-se. A "melhor" solução pode ser tanto a primeira solução ótima encontrada pela operação de movimentação da vizinhança, ou pode ser baseado na movimentação completa da vizinhança.

A fim de evitar ciclos, uma lista de "movimentos tabu" é empregada. Normalmente esta lista proíbe certos movimentos que levam à revisitação de uma solução previamente encontrada. Esta lista de movimentos tabu é atualizada conforme o algoritmo procede, de modo que um movimento adicionado à lista tabu só é removido após um certo número de iterações (o "posse tabu") ter passado. Mesmo não sendo obrigatórios, é comum a presença dos movimentos tabu, uma vez que eles conduzem a uma melhor solução [14].

Algoritmo 2: Pseudocódigo para o algoritmo TS
Gerar solução inicial Enquanto Gerar a vizinhança Avaliar a vizinhança Determinar a melhor solução da vizinhança Atualizar a lista tabu Intensificar e Diversificar Até que o critério de parada seja atendido

1.4.3. Simulated Annealing

Simulated Annealing (SA) foi proposto por Kirkpatrick [15] e é um algoritmo que realiza uma busca local de propósito geral, inspirada no processo de recozimento de materiais [15]. SA tem muito em comum com o TS, pois ambos examinam movimentos possíveis de uma única solução de partida. SA incorpora um componente estatístico em que movimentos para as piores soluções são aceitos com uma determinada probabilidade, que diminui ao longo do algoritmo.

Esta probabilidade está relacionada ao conceito de "temperatura". Mais precisamente, um movimento que piore a solução em função de Δ é aceita com uma probabilidade de $e^{-\Delta/T}$, onde T é a temperatura atual. Quanto maior a temperatura T, maior a probabilidade de a solução aceitar a mudança. Daí essa probabilidade diminui à medida que a temperatura diminui. Essa temperatura vai sendo reduzida durante o curso do algoritmo de acordo com um fator α ($0 < \alpha < 1$). [15]

Algoritmo 3: Pseudocódigo para o algoritmo SA
Gerar solução inicial S Gerar temperatura inicial T Enquanto (T > TFinal) S' = gera vizinho qualquer Se (*fitness (S) < fitness (S')*) então S = S' Senão S = S' com certa probabilidade $\boldsymbol{e^{-\Delta/T}}$, Fim do Se Atualiza temperatura Fim do enquanto Retorna S

1.4.4. Otimização por enxame de partículas

O algoritmo do Particle Swarm Optimization (PSO) foi proposto por James Kennedy e Russell Eberhart [16]. Nele, inicialmente é gerado aleatoriamente um

'enxame de partículas' (que representa o conjunto de possíveis soluções) num espaço de busca multidimensional.

Dentro desse espaço cada indivíduo da população é denominado partícula e cada elemento tem dois valores associados a ele: sua posição e sua velocidade. A posição representa o lugar onde o indivíduo estaria no espaço, ou seja, esse valor é modificado a cada geração (iteração). Já a velocidade é um valor resultante da influência de três vetores que se somam. Esses vetores são chamados de inércia, memória e comparação. A inércia ajuda a partícula a manter a direção que vinha seguindo. O segundo vetor, memória, atua na atração da partícula na direção da melhor posição até o momento, ou seja, a melhor posição ocupada pela partícula dentro da sua vida. Já a comparação atua semelhante à memória, só que atrai a partícula na direção do melhor ponto do espaço até o momento, descoberto pelo conjunto.

Dessa forma, cada partícula ajusta sua posição de acordo com sua experiência e a dos demais componentes da população. Com isso, o algoritmo do PSO combina uma de busca local com uma busca global. [11, 17]

Algoritmo 4: Pseudocódigo para o algoritmo PSO
Gerar uma população inicial de partículas: P Inicializar aleatoriamente a posição inicial (x) de cada partícula i de P. Atribua uma velocidade inicial (v) igual para todas as partículas. Para cada partícula i de P: Calcular sua aptidão $\boldsymbol{f_i}$ através da função fitness escolhida Calcular e melhor posição da partícula i até o momento $\boldsymbol{x_{ibest}}$ Selecionar a partícula com a melhor aptidão de toda a população ($\boldsymbol{g_{best}}$). Para cada partícula i em P: Atualize a velocidade da partícula pela fórmula: $v_i = wv_i + c_1 rand()(p_{id} - x_i) + c_2 Rand()(p_{gd} - x_i)$ Atualizar a posição da partícula pela fórmula: $x_i = x_i + v_i$ Se condição de término não for alcançada, retorne ao passo 4

2. Formulação do problema

2.1. Problema da seleção de portfólio

O problema de seleção de portfólio deve ser entendido como o estudo do investimento de um patrimônio. É um processo que mede a relação entre custo X benefício entre risco e o retorno esperado para encontrar o melhor portfólio de ativos no mercado de ações.

A composição de um portfólio varia muito devido ao fato de existirem diferentes tipos de investidores que desejam assumir diferentes riscos e retornos. Por exemplo, o risco assumido por uma pessoa depende de seus fatores pessoais, perspectivas futuras, e outros fatores. Diante disso, nunca existirá um portfólio de ações que agrade a todos, uma vez que quanto maior o retorno desejado maior será o risco a se assumir.

Markowitz (1952) contribuiu com o processo de seleção de portfólio de investimentos. Este desenvolveu uma metodologia de avaliação e compensação do risco através da diversificação de investimentos. Esta teoria desenvolvida por eles é muito utilizada nos dias hoje [18, 19, 20].

2.2. Modelo de Markowitz

Harry Markowitz [2, 21] desenvolveu um modelo para a seleção de portfólios. Nesta abordagem, assume-se que os retornos dos ativos seguem uma distribuição normal multivariada. Isto significa que o retorno de um portfólio de ações pode ser completamente descrita pelo retorno esperado e pela variância (risco). Diante disso, Markowitz propôs que deve haver um portfólio que maximiza o retorno esperado, minimizando a variância, e esta deve ser recomendada para um investidor [18].

No entanto, a solução proposta for Markowitz possui limitações, que podem ser de dois tipos [4]:

1) Apresenta leptocurtose em suas distribuições de probabilidade. Leptocurtose é o termo utilizado para descrever uma distribuição com densidade de probabilidade elevada em suas extremidades, gerando grande probabilidade de ocorrência de eventos extremos [22, 23].

2) Restrições de integridade que limitam um portfólio a ter um determinado número de ativos, ou impor limites sobre a proporção de um determinado ativo, não podem ser facilmente aplicados. Restrições deste tipo são de importância prática [18].

2.2.1. Problema sem restrição

O modelo seguido por esse estudo segue a abordagem de Chang e colaboradores [4]. Seja:

- N o número de ações disponíveis
- μ_i o retorno esperado para a ação i $(i = 1, \dots, N)$
- σ_{ij} a covariância entre os ativos i e j $(i = 1, \dots, N\ ;\ i = 1, \dots, N)$
- R^* o retorno desejado esperado

E as variáveis de decisão:

- w_i representa a proporção $(0 \leq w_i \leq 1)$ para a ação i $(i = 1, \dots, N)$

Utilizando essas variáveis e o modelo padrão de variância média de Markowitz (Eqs. 1 - 6), tem-se a descrição do problema de otimização de portfólio sem restrições.

Eq. 1: minimizar $\sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j \sigma_{ij}$

Sujeito a:

Eq. 2: $\sum_{i=1}^{N} w_i \mu_i = R^*$

Eq. 3: $\sum_{i=1}^{N} w_i = 1$

Eq. 4: $0 \leq w_i \leq 1, \quad (i = 1, \dots, N)$

Eq. (1) minimiza a variância total (risco) associado com o portfólio, enquanto Eq. (2) garante que o portfólio tem um retorno esperado de R^*. Eq. (3) assegura que as proporções adicionam 1.

Esse conjunto de fórmulas (Eqs. 1 - 4) é um simples problema de programação não linear (quadrático) para os quais já existem algoritmos computacionalmente efetivos para a sua resolução. Por conta disso, na prática, calcular o ótimo para esses problemas não apresenta muita dificuldade para um conjunto de dados específico.

Uma observação acerca dessas equações (1 - 4) é que elas podem ser expressas em relação à correlação ρ_{ij} entre as ações i e j $(-1 \leq \rho_{ij} \leq 1)$ e os desvios padrões s_i, s_j no retorno desses ativos desde que $\sigma_{ij} = \rho_{ij} s_i s_j$

2.2.2. Fronteira eficiente

Ao resolver o conjunto de equações acima (Eqs. 1 - 4) para diferentes valores de R^* podemos traçar a fronteira eficiente. Que consiste numa curva não decrescente suave que dá a melhor relação custo X benefício entre o risco e o retorno, isto é, a curva representa o conjunto de portfólios ótimos, também denominadas de portfólios ótimos de Pareto.

Uma possível FE é mostrada na figura 1 para ativos extraídos do índice FTSE 100, ação presente no mercado financeiro do Reino Unido.

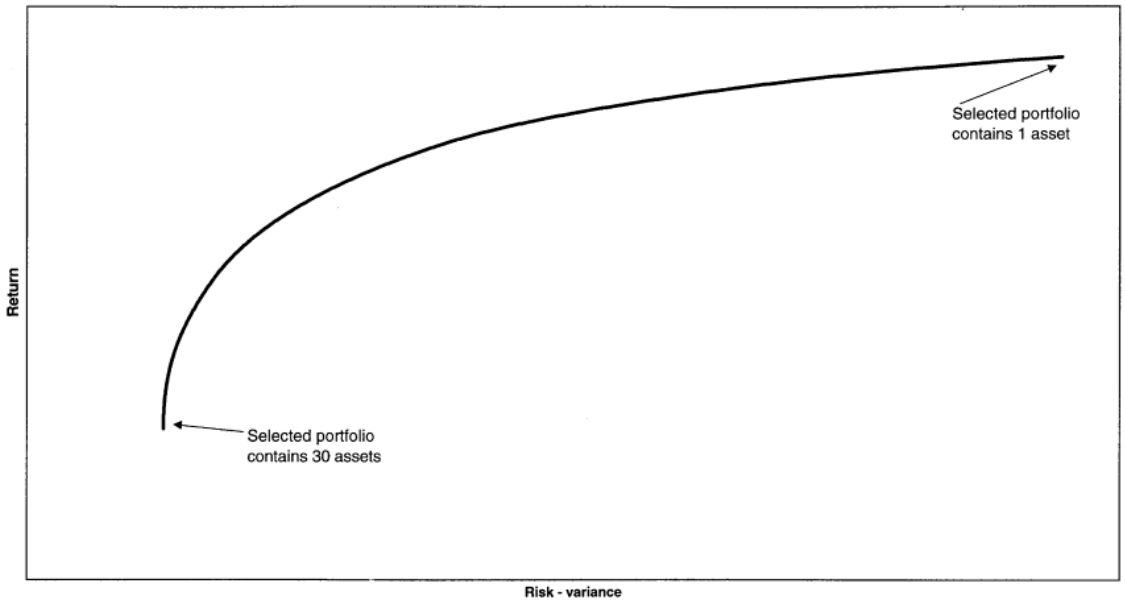

Figura 1. Fronteira eficiente relativa à ação FTSE 100 [4]

Para o caso do modelo sem restrição, é prática padrão para traçar a fronteira eficiente introduzir um parâmetro de ponderação $\lambda\,(0 \leq \lambda \leq 1)$ que é calculado através das equações abaixo (5 - 7):

Eq. 5: $minimizar\ \lambda\left[\sum_{i=1}^{N}\sum_{j=1}^{N} w_i\, w_j \sigma_{ij}\right] - (1 - \lambda)\left[\sum_{i=1}^{N} w_i \mu_i\right]$

Sujeito a:

Eq. 6: $\sum_{i=1}^{N} w_i = 1$

Eq. 7: $0 \leq w_i \leq 1, \quad (i = 1, \ldots, N)$

Na equação (5), a solução em que λ = 0 representa o retorno esperado máximo (independentemente do risco envolvido) e a solução ideal envolverá apenas o único ativo com o maior retorno. O caso em que λ = 1, representa o valor mínimo do risco (independentemente do retorno envolvido) e, nesse caso, a solução ótima tipicamente envolverá mais de uma ação para o portfólio de Pareto. E os casos em que $0 < \lambda < 1$ representam a relação custo X benefício entre o risco e o retorno, gerando soluções entre os dois extremos λ = 0 e λ = 1.

Semelhante às igualdades anteriores (1 - 4), resolvendo estas equações quadráticas (5 - 7), para diferentes valores de λ, pode-se traçar a fronteira de eficiência. Por exemplo, considerando $\lambda = 0{,}25$: então, o objetivo (eq. 5) que é minimizar, seria 0,25 [risco] – 0,75 [retorno]. Considerando a figura 1, que mostra a curva da fronteira de eficiência para diferentes valores de R^*, pode-se traçar uma série de novas linhas com a mesma eficiência 0,25 [risco] – 0,75 [retorno] = Z e escolher o menor valor de Z. Reorganizando a equação, [retorno] = (1/3)[risco] – (3/4)Z, ou seja, linhas de inclinação (1/3) e interseção do eixo y em – 3/4 de Z. Minimizar Z corresponde, portanto, à maximizar o retorno (interseção com o eixo y)

Assim, variando λ em relação às equações (5 – 7), pode-se traçar exatamente a mesma curva de fronteira eficiente que seria obtida utilizando as equações (1 – 4), para diferentes valores de R^*.

2.3. Modelo CCMV

O modelo padrão de Markowitz não contém qualquer cardinalidade ou limite de restrição, o que restringem o número de ativos e, os limites superiores e inferiores da proporção de cada ativo no portfólio, respectivamente. Chang e colaboradores [4] e Fernandez e Gomez [24] usaram uma versão modificada do modelo de Markowitz chamado de Modelo de restrição da cardinalidade da variância média (CCMV). Este modelo também foi utilizado neste estudo.

2.3.1. Modelo com restrição:

O modelo seguido por esse estudo segue a abordagem de Chang e colaboradores [4]. Seja:

- K número desejado de ações no portfólio
- ε_i a proporção mínima de i caso alguma ação de i seja mantida.
- δ_i a proporção máxima de i caso alguma ação de i seja mantida.
- Onde, $0 \leq \varepsilon_i \leq \delta_i \leq 1$ $(i = 1, \dots, N)$

Na prática, ε_i representa o nível mínimo de transações para o ativo i e δ_i limita a exposição do portfólio ao ativo i. Para melhores resultados, é introduzida uma variável de decisão zero-um.

- $z_i = 1$, Caso alguma ação i $(i = 1, \dots, N)$ seja mantida.
- $z_i = 0$, caso contrário

Diante disso, o problema de restrição da cardinalidade para problemas de portfólio consiste em (Eqs. 8 - 13):

Eq. 8: $minimizar \sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j \sigma_{ij}$

Sujeito a:

Eq. 9: $\sum_{i=1}^{N} w_i \mu_i = R^*$

Eq. 10: $\sum_{i=1}^{N} w_i = 1$

Eq. 11: $\sum_{i=1}^{N} z_i = K$

Eq. 12: $\varepsilon_i z_i \leq w_i \leq \delta_i z_i$

Eq. 13: $z_i \in [0,1], \quad i = 1, \ldots, N$

A equação 8 minimiza a variância total (risco) associada com o portfólio enquanto a equação 9 garante que o portfólio tenha um retorno esperado de R^*. A equação 10, garante que seja adicionado 1 em relação a proporção. Já a equação 11 garante que exatamente K ações sejam mantidas. Equação 12 assegura que se algum ativo i for mantido ($z_i = 1$) sua proporção deve ficar entre ε_i e δ_i, por outro lado se nenhuma ação i for mantida ($z_i = 0$) sua proporção é zero. Equação (13) é a restrição de integridade.

Outro ponto importante é que este problema foi formulado com uma igualdade (ao invés de uma desigualdade) em relação ao número de ativos no portfólio (Eq. (11)). O objetivo dessa escolha é que caso se resolva a igualdade restrita, em seguida, qualquer situação que envolva as desigualdades (limites inferiores ou superiores sobre o número de ativos no portfólio) pode ser facilmente tratada.

Por exemplo: $K_L \leq \sum_{i=1}^{N} z_i \leq K_U$. Nesse exemplo, caso o número de ativos no portfólio se encontre entre os valores de K_L e K_U ($K_L \neq K_U$), então esta função pode ser tratada pelas Eqs. (8 - 13) para todos os valores diferentes de K (Eq. 11), entre K_L e K_U, isto é, K = K_L, $K_L + 1$, ..., K_U.

Mesmo parecendo complicada, nota-se que esta abordagem tem o mérito. Uma vez que, na prática, a decisão quanto ao número de ativos (K) para ter no portfólio escolhido é aquele que só pode ser decidida pelo tomador de decisão à luz da relação custo X benefício entre os três fatores envolvidos (risco, retorno, K).

Uma observação sobre a Eqs. (8 – 13) é que estas são problemas de programação não linear. Em contraste com as equações sem restrição (1 - 4), não existem algoritmos computacionais eficientes para aquelas. Em termos gerais, para a resolução desse tipo de problemas utilizam-se algoritmos disponíveis para resolver problemas não lineares [19, 20].

3. Solução proposta

Num ambiente real, as abelhas comuns, assumem diferentes funções dentro de suas colônias ao longo do tempo. Uma colmeia típica pode ter de 5.000 a 20.000 abelhas, dentre elas podem-se observar que as abelhas adultas geralmente se tornam operárias. As abelhas operárias normalmente se enquadram em uma destas três funções: ativas, protetoras e inativas.

As abelhas operárias ativas viajam até fontes de alimentos, examinam as fontes de alimentos da vizinhança, coletam alimentos e retornam para a colmeia.

As abelhas protetoras investigam a área ao redor da colmeia, em geral, uma região de até 130 quilômetros quadrados, procurando por novas fontes de alimentos atraentes. Aproximadamente 10% das abelhas operárias de uma colmeia são responsáveis pela proteção.

A todo o momento, algumas abelhas operárias ficam inativas. Essas operárias inativas aguardam próximas à entrada da colmeia. Quando as operárias ativas e as protetoras retornam à colmeia, dependendo da qualidade da fonte de alimento que acabaram de visitar, elas podem fazer uma "dança agitada" para as abelhas inativas que estavam aguardando. Existem fortes evidências de que essa dança transmite informações às abelhas inativas sobre o local e a qualidade da fonte de alimento. As operárias inativas recebem essas informações sobre a fonte de alimento a partir da dança e podem se tornar operárias ativas.

Em geral, uma abelha operária ativa continua coletando alimentos de uma fonte específica até que a fonte se esgote e, nesse momento, a abelha se torna uma operária inativa. [7, 25].

3.1. Algoritmo de Colônia Artificial de Abelhas:

A Colônia Artificial de Abelhas (CAA) foi proposta por Karaboga em 2005 [8]. Este é um algoritmo de inteligência coletiva baseada no comportamento de uma colônia de abelhas (descrito acima). Assim como todos os algoritmos de inteligência coletiva, o

CAA combina um processo de exploração individual com experiências coletivas passadas [25].

No algoritmo CAA, o comportamento das abelhas baseia-se num modelo matemático baseado em três componentes essenciais: abelhas operárias, abelhas não operárias (seguidoras ou exploradoras) e fontes de alimento. Estas representam as possíveis soluções do problema, enquanto as abelhas são os agentes de busca da solução.

Um ponto importante a ser destacado desse algoritmo consiste na execução dos passos *forward* e *backward* praticados pelas abelhas a cada iteração. No passo *forward*, as abelhas voam da colmeia e escolhem algumas soluções do conjunto de soluções parciais. Já no passo *backward*, todas as abelhas voltam para a colmeia. Esse reagrupamento permite que todas as abelhas troquem, com as demais, informações sobre a qualidade das soluções parciais criadas e decida se abandona sua solução e torna-se seguidora novamente, se continua a seguir a mesma solução parcial ou se recruta as companheiras antes de retornar a solução criada. [26, 27]. Esses passos são repetidos durante as iterações do algoritmo até que algum dos critérios de parada (por exemplo, número máximo de ciclos) seja atingido. O algoritmo pode ser expresso pela imagem a seguir (figura 2).

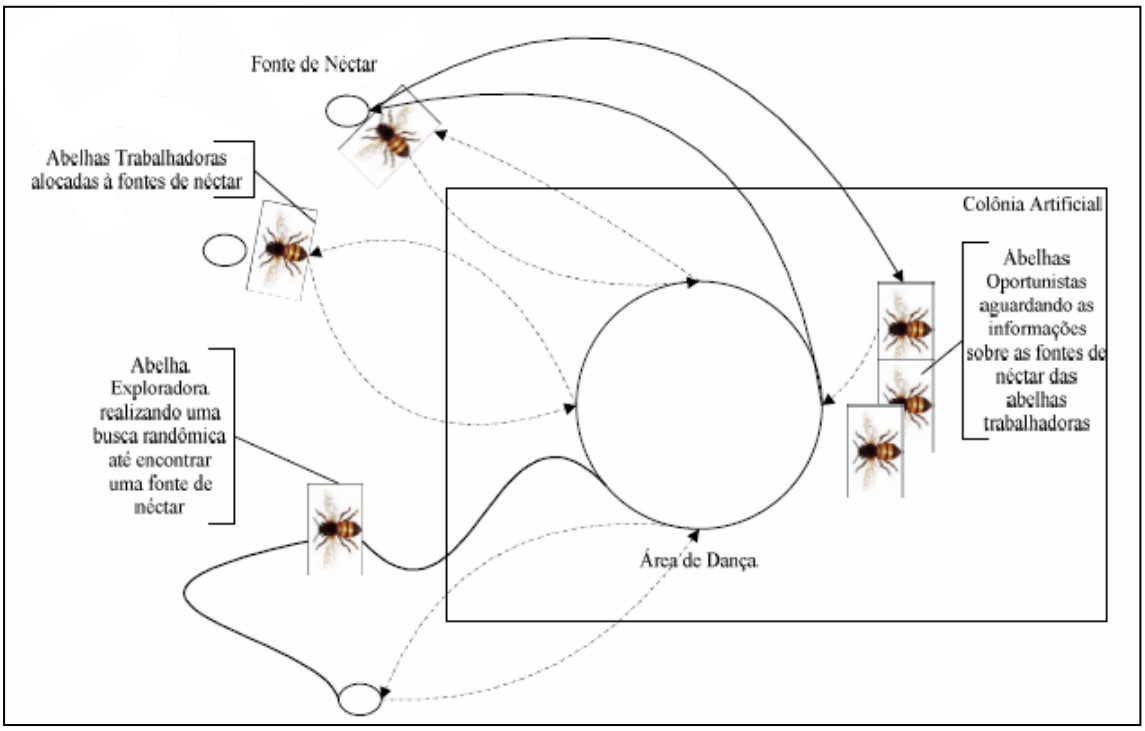

Figura 2 Representação do algoritmo CAA [28]

Abaixo segue o pseudocódigo contendo a descrição do algoritmo CAA [26]:

Algoritmo 5: Pseudocódigo para o algoritmo CAA
Inicialize a população com soluções aleatórias Avaliar a aptidão (fitness) da população Enquanto (número máximo de ciclos não for atingido) / / Formação de nova população. Selecionar fontes de alimentos na busca da vizinhança Recrutar as abelhas seguidoras para irem às fontes de alimentos selecionadas e avaliar suas novas aptidões. Selecionar as abelhas com melhores fitnesses Com as demais abelhas (exploradoras), ainda não selecionadas, fazer busca aleatório e avaliar suas aptidões. Fim

3.2. CAA para solução do CCMV

Considerando que cada abelha é um vetor N dimensional e que cada dimensão representa uma ação. Cada abelha pode ser descrita pelas variáveis x_{ij} $(i = 1, \dots, N; j = 1, \dots, N)$ e z_{ij} $(i = 1, \dots, SN;\ j = 1, \dots, N)$, onde SN é o número de abelhas empregadas. Assim, o número de dimensões que uma abelha possui é 2 × N. Pode-se observar que as abelhas representam soluções candidatas, e cada abelha deve satisfazer as restrições das equações (Eqs. 8 - 13). Baseado na abordagem de Karaboga [29, 30] e utilizando as restrições propostas por Tunchan [5], o algoritmo de arranjo é apresentado para garantir a viabilidade das soluções apresentadas a seguir.

Algoritmo 6: Organizar abelha

Seja abelha (i):

Q: conjunto de $\boldsymbol{K_i^*}$ ativos distintos de i;

$\mathbf{z_{ij}}$: o valor atual de decisão para o ativo j;

$\boldsymbol{x_{ij}}$: o valor atual da proporção para o ativo j;

Enquanto ($\boldsymbol{K_i^*} < K$):

Se aleatório [0,1] < 0.5 então j = um ativo selecionado aleatoriamente de tal forma que $\boldsymbol{j} \notin \boldsymbol{Q}$

Se não j = selecione o ativo com maior valor $\boldsymbol{c_j}$ de tal forma que $\boldsymbol{j} \notin \boldsymbol{Q}$

Fim do se

$\boldsymbol{z_{ij}}$ = 1, Q = Q ∪ [**j**], $\mathbf{K_i^*} = \mathbf{K_i^*} + 1$

Fim do enquanto

Enquanto ($\boldsymbol{K_i^*} > K$):

Se aleatório [0,1] < 0.5 então j = um ativo selecionado aleatoriamente de tal forma que $\boldsymbol{j} \in \boldsymbol{Q}$

Se não j = selecione o ativo com menor valor $\boldsymbol{c_j}$ de tal forma que $\boldsymbol{j} \in \boldsymbol{Q}$

Fim do se

$\boldsymbol{z_{ij}}$ = 0, Q = Q −[**j**], $\mathbf{K_i^*} = \mathbf{K_i^*} - 1$

Fim do enquanto

Enquanto (critério de parada não é atingido)

.$\boldsymbol{\chi} = \sum_{j \in Q} \boldsymbol{x_{ij}}$; $\boldsymbol{x_{ij}} = \boldsymbol{x_{ij}}/\boldsymbol{\chi}$;

.$\boldsymbol{\eta} = \sum_{j \in Q} \mathbf{Max}(\mathbf{0}, \boldsymbol{x_{ij}} - \boldsymbol{\delta_j})$; $\boldsymbol{\phi} = \sum_{j \in Q} \mathbf{Max}(\mathbf{0}, \boldsymbol{\varepsilon_j} - \boldsymbol{x_{ij}})$

Se $\boldsymbol{\eta} = \mathbf{0}\ \boldsymbol{e}\ \boldsymbol{\phi} = \mathbf{0}$ então Sair do algoritmo do algoritmo de organização

Para j = 1 até N

Se $\boldsymbol{z_{ij}}$ = 1 então

Se $\boldsymbol{x_{ij}} > \boldsymbol{\delta_i}$ então $\boldsymbol{x_{ij}} = \boldsymbol{\delta_j}$

Fim do Se

Se $\boldsymbol{x_{ij}} < \boldsymbol{\varepsilon_i}$ então $\boldsymbol{x_{ij}} = \boldsymbol{\varepsilon_j}$

Fim do Se

Fim do Se

Fim do para

Fim do enquanto

Admitindo Q como o conjunto de ativos mantidos pela abelha i, e cada abelha i possuindo K^* ativos distintos, onde $K^* = \sum_{j=1}^{N} z_{ij}$. Se $K^* < K$, alguma ação deve ser adicionada à Q; Se $K^* > K$, alguma ação deve ser removida de Q até que $K^* = K$. Considerando que $K^* < K$: se um número aleatório no intervalo [0,1] for inferior a 0,5, será selecionado um ativo que não está em Q e adicioná-lo à Q; caso contrário, será selecionada a ação que apresentar o maior valor de c_i que não está em Q e adicioná-lo à Q. O valor c_i é calculado como sendo a proporção entre o retorno médio e o risco médio, que pode ser calculado através da equação (14) [5].

Eq. 14: $$c_i = \frac{\theta_i + \theta}{\rho_i + \rho}, \qquad i = 1, \dots, N$$

Onde: $\theta_i = 1 + (1 + \lambda)\mu_i$; $\theta = -\min(0, \theta_1, \dots, \theta_N)$;

$$\rho_i = {1 + \lambda \sum_{j=1}^{N} \sigma_{ij}}\Big/{N}; \rho = -\min(0, \rho_1, \dots, \rho_N);$$

Quando $K^* > K$, é selecionado uma ação de Q e esta é removida ou então é selecionada a ação com menor valor c_i e esta será removida de Q.

Para a restrição $\sum_{i=1}^{N} x_i = 1$, fazer $\chi = \sum_{j \in Q} x_{ij}$ e $x_{ij} = x_{ij}/\chi$ para todo $j \in Q$. Devido às restrições $\varepsilon_i z_i \leq w_i \leq \delta_i z_i$, $i = 1, \dots, N$, se $x_{ij} > \delta_j$ então $\boldsymbol{x_{ij} = \delta_i}$; Se $\boldsymbol{x_{ij} < \varepsilon_i}$ então $\boldsymbol{x_{ij} = \varepsilon_j}$.

3.2.1. Função Fitness (aptidão)

A função de aptidão utilizada, nesse estudo, foi proposta por Karaboga e Basturk [29, 30], e consiste em (Eq. 15):

Eq. 15: $$fitness_i = \begin{Bmatrix} 1/(1 + f_i), \ caso\ f_i \geq 0 \\ 1 + abs(f_i), \ caso\ f_i < 0 \end{Bmatrix}$$

Onde f_i é o valor de custo da solução representado pela equação abaixo (Eq. 16):

Eq. 16: $$f_i = \lambda \left[\sum_{j=1}^{N} \sum_{k=1}^{N} z_{ij} x_{ij} z_{ik} x_{ik} \sigma_{jk} \right] - (1 - \lambda) \left[\sum_{j=1}^{N} z_{ij} x_{ij} \mu_j \right]$$

3.2.2. Produção inicial das fontes de comida

As fontes iniciais de alimentos são produzidas aleatoriamente dentro dos limites dos parâmetros, através das equações a seguir: (17 e 18) [29, 30].

Eq. 17: $$z_{ij} = round \left(\frac{1}{1 + e^{-z_j^{min} + rand(0,1)\,(z_j^{max} - z_j^{min})}} - 0{,}06 \right)$$

Eq. 18: $$x_{ij} = x_j^{min} + rand\,(0,1)(x_j^{max} - x_j^{min})$$

Após a inicialização, as fontes de alimento (soluções) são submetidas a repetição dos ciclos do processo de busca de alimento das abelhas empregadas, das abelhas exploradoras e das abelhas seguidoras. Um dos critérios para o fim dessa busca é o atingimento do número máximo ciclos.

3.2.3. Mandando abelha empregada para fonte de alimento

Cada abelha empregada está relacionada com uma fonte de alimento. Logo existirão tantas fontes de alimento quantas forem às empregadas. A cada ciclo, uma abelha empregada encontra uma fonte de alimento vizinho e avalia a sua qualidade, através do cálculo de aptidão. O cálculo para as abelhas encontrarem uma fonte de alimento vizinho é definido pelas equações (19 e 20) [29, 30]:

Eq. 19: $$z'_{ij} = round \left(\frac{1}{1 + e^{-z_{ij} + \phi_{ij}\,(z_{ij} - z_{lj})}} - 0{,}06 \right)$$

Eq. 20: $$x'_{ij} = \begin{Bmatrix} x_{ij} + \phi_{ij}\,(x_{ij} - x_{lj}), se\ z'_{ij} = 1 \\ x_{ij}, \quad caso\ contrário \end{Bmatrix}$$

Onde j é um número inteiro aleatório no intervalo {1,2, ..., N} e l é um índice escolhido aleatoriamente no conjunto {1, 2, ..., SN} que tem que ser diferente de i. E Φ_{ij} é um número real aleatoriamente escolhido no intervalo de [-1, 1].

Diante disso, é feita uma comparação entre a nova fonte de alimento (z_i, x_i) e a antiga (z_i', x_i'); então o melhor resultado é selecionado dependendo do valor da função fitness calculada (Eq. 15). Se o valor obtido em (z_i', x_i') for superior ao valor (z_i, x_i), a abelha memoriza a posição nova e abandona o antigo, caso contrário, a posição anterior é mantida na memória (não sendo mantido um histórico da memória da abelha). O objetivo dessa operação consiste em fazer com que a abelha, a cada iteração, tenda para a melhor solução. Se (z_i, x_i) não pode ser melhorado, o seu contador de quantidade de exploração é incrementado de 1, caso contrário, o contador é reposto a 0.

3.2.4. Abelhas seguidoras

Depois de todas as abelhas empregadas completarem suas pesquisas, elas compartilham suas informações sobre suas fontes de alimento com as abelhas seguidoras. Então, cada abelha seguidora avalia as informações e escolhe uma fonte de alimento, determinado pela probabilidade. Esse valor é conhecido como probabilidade de recrutamento, pois representa a probabilidade de a abelha empregada recrutar a abelha seguidora para a sua fonte de alimento. Uma característica importante dessa etapa é que se utiliza o esquema de seleção Roleta, descrito pela equação (21) [7, 29 , 30].

Eq. 21: $p_i = \frac{fitness_i}{\sum_{i=1}^{SN} fitness_i}$

Se o valor da probabilidade p_i for maior que um número real aleatório no intervalo [0, 1], então a abelha seguidora escolhe a fonte de alimento através das equações (19 e 20). Logo, a comparação entre as fontes de alimento é utilizada e a abelha seguidora, semelhante às abelhas empregadas, ou memoriza a posição nova e esquece a antiga ou a mantém a posição anterior. Se (z_i, x_i) não pode ser melhorado, o seu contador de quantidade de exploração é incrementado de 1, caso contrário, o contador é reposto a 0.

3.2.5. Critério de abandono

Depois de todas as abelhas, empregadas e exploradoras, completarem suas buscas, se uma posição não pode ser melhorada através de um "limite" (limite de visitas), a fonte de alimento é abandonada. Esta pode ser substituída por novas fontes de alimento criadas usando as Eq. 17 e 18 [28, 29, 30].

O funcionamento da Colmeia pode ser descrita da seguinte forma.

Algoritmo 7: CAA para problema de portfólio

Inicializar $\lambda = 0$

Enquanto ($\boldsymbol{\lambda} \leq \mathbf{1}$)

Inicializar a população de soluções $\boldsymbol{z_{ij}}, \boldsymbol{x_{ij}}, \boldsymbol{i} = \mathbf{1}, \dots, \boldsymbol{SN}; \boldsymbol{j} = \mathbf{1}, \dots, \boldsymbol{N}$ (Eq. 17 e 18),

$\boldsymbol{limite_i} = \mathbf{0}$, onde $\boldsymbol{limite_i}$ é número que é utilizado para o abandono

Organizar abelha (i), para $\boldsymbol{i} = \mathbf{1}, \dots, \boldsymbol{SN}$; (Algoritmo 6)

Calcular o grau de aptidão das abelhas: $\boldsymbol{fitness_i}, \boldsymbol{para\ i} = \mathbf{1}, \dots, \boldsymbol{SN}$ (Eq. 15)

Para contador = 1até $\lfloor \mathbf{1000}\boldsymbol{N}/\boldsymbol{SN} \rfloor$

// Fase das abelhas empregadas

Para i = 1 até SN

$$.\boldsymbol{z'_{ij}} = \boldsymbol{round}\left(\frac{\mathbf{1}}{\mathbf{1} + \boldsymbol{e}^{-z_{ij}+\phi_{ij}\,(z_{ij}-z_{lj})}} - \mathbf{0,06}\right)$$

Se $\boldsymbol{z'_{ij}}$ = 1

Então Produzir nova fonte de alimento $\boldsymbol{x'_{ij}}$ para abelha (i) utilizando a Eq. 20

Fim do Se

Organizar abelha (i)

Calcular o grau de aptidão das abelhas: $\boldsymbol{f'_i}$

Se $\boldsymbol{f'_i} < \boldsymbol{f_i}$ Então $\boldsymbol{z_{ij}} = \boldsymbol{z'_{ij}}$; $\boldsymbol{x_{ij}} = \boldsymbol{x'_{ij}}$

Se não $\boldsymbol{limite_i} = \boldsymbol{limite_i} + \mathbf{1}$

Fim do Se

Fim do Para

Calcular os valores de probabilidade de $\boldsymbol{p_i}$ (Eq. 21).

// Fase das abelhas seguidoras

totalAbelhas = 0; i = 1

Enquanto

Se aleatório [0,1] < $\boldsymbol{p_i}$,

Então $\boldsymbol{z'_{ij} = round\left(\frac{1}{1+e^{-z_{ij}+\phi_{ij}(z_{ij}-z_{lj})}} - 0,06\right)}$

Se $\boldsymbol{z'_{ij}}$ = 1

Então Produzir nova fonte de alimento $\boldsymbol{x'_{ij}}$ para abelha (i) utilizando a Eq. 20

Fim do Se

Organizar abelha (i)

Calcular o grau de aptidão das abelhas: $\boldsymbol{f'_i}$

Se $\boldsymbol{f'_i < f_i}$ Então $\boldsymbol{z_{ij} = z'_{ij}}$; $\boldsymbol{x_{ij} = x'_{ij}}$

Se não $\boldsymbol{limite_i = limite_i + 1}$

Fim do Se

totalAbelhas = totalAbelhas + 1

Fim do Se

Até (totalAbelhas = SN)

// Fase das abelhas exploradoras

Gerar uma solução aleatoriamente nova (abelha exploradora), substituindo alguma abelha antiga.

Memorize a melhor solução alcançada até agora

Fim do Para

.$\boldsymbol{\lambda = \lambda + \Delta\lambda}$

Fim do enquanto

A partir do algoritmo descrito acima, observa-se que o processamento das abelhas trabalhadoras (que estão em maior quantidade) é a atividade que demanda mais tempo na execução do algoritmo. Diante disso, não existe nenhum atalho para acelerar o andamento do CAA proposto. Além disso, o critério de parada utilizando (objetivando a comparação com Tunchan [5]) foi o de $\boldsymbol{\lambda \leq 1}$, não sendo levada em consideração a iteração em que a melhor solução foi encontrada.

4. Experimentos Computacionais

Utilizamos os parâmetros e atributos definidos por Tunchan [5], com o objetivo de criar um mesmo ambiente e, consequentemente, fazer as devidas comparações.

Diante disso, os dados de teste são referentes aos preços semanais das ações: Hang Seng, de Hong Kong, DAX 100 da Alemanha, FTSE 100 do Reino Unido, S&P 100 dos EUA e Nikkei 225 do Japão, entre março de 1992 e setembro de 1997. Para cada conjunto de dados de teste, o número, N, de ativos diferentes é de 31, 85, 89, 98 e 225, respectivamente. Os dados de teste, que também foram utilizados por Tunchan [5], foram obtidos em http://people.brunel.ac.uk/%7Emastjjb/jeb/orlib/portinfo.html.

Todos os resultados foram computados usando os valores k = 10, $\varepsilon_i = 0{,}01$ $(i = 1, \dots, N)$ e $\delta_i = 1$ $(i = 1, \dots, N)$, na formulação do problema, e $\Delta\lambda = 0{,}02$ para a implementação do algoritmo. O número de diferentes λ é igual 51, denominado ξ.

Assumindo o conjunto de portfólios ótimos de Pareto, obtidos por cada algoritmo, podemos traçar suas fronteiras de eficiência heurísticas. A fim de fazer comparação entre fronteiras de eficiência padrões com as correspondentes fronteiras heurísticas utilizou-se a distância euclidiana, erro da variância e erro de retorno. A fronteira padrão é calculada usando 2000 valores diferentes para resolução da equação (Eq. 5).

Fazendo o par (v_i^s, r_i^s) $(i = 1, \dots, 2000)$ representar a variância e o retorno a partir do ponto na fronteira de eficiência padrão. E o par $\left(v_i^h, r_i^h\right)$ $(i = 1, \dots, \xi)$ representando a variância e o retorno da fronteira heurística. Então, para descobrimos a menor distancia entre (v_i^s, r_i^s) e $\left(v_i^h, r_i^h\right)$, utilizamos o cálculo da equação: $i_j = \min_{i=1,\dots,2000}\left(\sqrt{\left(v_i^s - v_j^h\right)^2 + \left(r_i^s - r_j^h\right)^2}\right)$ $j = 1, \dots, \xi$.

A partir daí, foi definido: [5]

- A média da distância euclidiana:

$$\left(\sum_{j=1}^{\xi}\sqrt{\left(v_i^s - v_j^h\right)^2 + \left(r_i^s - r_j^h\right)^2}\right) \times {}^{1}/_{\xi}$$

- Erro da variância (%):

$$\left(\sum_{j=1}^{\xi} 100 \mid v_{ij}^s - v_j^h \mid / v_j^h\right) \times {}^{1}/_{\xi}$$

- Erro do retono (%):

$$\left(\sum_{j=1}^{\xi} 100 \mid r_{ij}^s - r_j^h \mid / r_j^h\right) \times {}^{1}/_{\xi}$$

Tabela 1. Resultados experimentais das heurísticas

Índice	**Quantidade de ações**		**GA**	**TS**	**SA**	**PSO**	**CAA**
HangSeng	31	Média de distancia euclidiana	0,004	0,004	0,004	0,0049	0,0000819
		Erro da variância (%)	1,6441	1,6578	1,6628	2,2421	1,4805
		Erro do retorno (%)	0,6072	0,6107	0,6238	0,7427	0,5372
DAX100	85	Média de distancia euclidiana	0,0076	0,0082	0,0078	0,009	0,000159
		Erro da variância (%)	1,6441	1,6578	1,6628	2,2421	8,139
		Erro do retorno (%)	0,6072	0,6107	0,6238	0,7427	1,3511

FTSE100	89	Média de distancia euclidiana	0,002	0,0021	0,0021	0,0022	0,000159
		Erro da variância (%)	2,866	4,0123	3,8205	3,0596	2,7629
		Erro do retorno (%)	0,3277	0,3298	0,3304	0,364	0,3114
S&P100	98	Média de distancia euclidiana	0,0041	0,0041	0,0041	0,0052	0,0000837
		Erro da variância (%)	3,4802	5,7139	5,4247	3,9136	3,2092
		Erro do retorno (%)	1,2258	0,7125	0,8416	1,404	0,6844
Nikkei	225	Média de distancia euclidiana	0,0093	0,001	0,001	0,0019	0,000006844
		Erro da variância (%)	1,2056	1,2431	1,2017	2,4274	3,8087
		Erro do retorno (%)	5,3266	0,4207	0,4126	0,7997	0,9802

A tabela 1 mostra os resultados obtidos com o CAA comparados com os valores obtidos pelos outros quatro algoritmos heurísticos. Os resultados de AG, TS, SA e PSO são obtidos a partir do estudo de Tunchan [5].

A partir dos valores observados na tabela 1 fica claro que a média de distância euclidiana no Algoritmo das Abelhas é menor do que nas outras quatro heurísticas, para esses cinco índices de mercado. Além disso, para HangSeng, FTSE 100 e S&P 100, CAA é o melhor algoritmo para resolver o modelo CCMV, capaz de aproximar a fronteira de eficiência padrão com o menor erro de variância e menor erro de retorno.

Uma observação acerca dos resultados obtidos é que as métricas utilizadas foram definidas para que os resultados pudessem ser comparados aos resultados obtidos por Tunchan [5]. Não sendo objetivo de esse estudo utilizar outras métricas de comparação ou até mesmo desempenho dos algoritmos como medidas de comparação.

Os resultados experimentais mostram que o algoritmo CAA pode ser utilizado para resolver o modelo CCMV no problema de otimização de portfólios e fornece soluções melhores do que outros algoritmos heurísticos. Por conta disso, economicamente, o algoritmo CAA dá melhores estratégias de portfólio do que outros métodos heurísticos para o investidor.

5. Conclusão

Este estudo foi focado em resolver o problema de seleção de portfólio, calculando sua fronteira eficiente através do algoritmo do CAA. O modelo utilizado para a seleção da portfólio foi CCMV, que representa uma variação da abordagem padrão Markowitz incluindo restrições de cardinalidade e de limitação de ativos no portfólio. Os resultados obtidos foram comparados com os obtidos por métodos heurísticos baseados em algoritmos genéticos, busca tabu, de recozimento simulado e PSO.

Os resultados experimentais mostraram que nenhum das cinco heurísticas claramente superou as outras em todos os tipos de investimento. No entanto, o Algoritmo CAA apresentou um conjunto de soluções com maior qualidade que os demais algoritmos. Diante disso, o algoritmo CAA apresentado neste trabalho é viável para resolver o problema de otimização de portfólio com restrições de cardinalidade.

Como trabalhos futuros fica a temática de tentar utilizar o CAA para ações do mercado brasileiro atual ou então buscar a comparação dos resultados obtidos com outros algoritmos como, por exemplo, o Ant Colony. [31].

6. Referências

[1] Jinchuan k, Can Z. Study on the optimization of portfolio based on entropy theory and mean-variance model. Service Operations and Logistics, and Informatics, 2008. IEEE/SOLI 2008. IEEE International Conference , vol.2, pp.2668-2672, 12-15 Oct. 2008.

[2] Markowitz H, Portfolio selection. The Journal of Finance. 1952; 7:77-91.

[3] Yuxi J, Suyan H, Xingsi L. A Maximum Entropy Model for Large-Scale Portfolio Optimization. Risk Management & Engineering Management, 2008. ICRMEM '08. International Conference, 610-615, 4-6 Nov. 2008.

[4] Tien-Jye C, Meade N, Beasley J, Sharaiha YM. Heuristics for cardinality constrained portfolio optimization. Computers & Operations Research 27 (2000); 1271-1302.

[5] Tunchan C. Particle swarm optimization approach to portfolio optimization, Nonlinear Analysis: Real World Applications. Volume 10, Issue 4, August 2009, Pages 2396-2406, ISSN 1468-1218, 10.1016/j.nonrwa.2008.04.023.

[6] Roudier F. Portfolio Optimization and Genetic Algorithms. Swiss Federal Institute of Technology (ETH) Zurich (2007).

[7] Teodorović D, Dell'Orco M. Bee colony optimization – a cooperative learning approach to complex transportation problems. University of Belgrade, Serbia and Montenegro. ACM Transactions on Computational Logic, 2011; 1–20.

[8] Hamid RG, Mehrshad F. Constrained Portfolio Selection using Particle Swarm Optimization, Expert Systems with Applications. Volume 38, Issue 7, July 2011, Pages 8327-8335, ISSN 0957-4174, 10.1016/j.eswa.2011.01.020.

[9] Suren, Wadhwa, and Terje LK. Comparison and Application of Metaheuristic Population-Based Optimization Algorithms in Manufacturing Automation. International Journal of Computer Science Issues 3rd ser. 8.6 (2011).

[10] Lejbman AG. Inteligência De Enxame E O Algoritmo Das Abelhas. Universidade De São Paulo, Brasil; 2009.

[11] Gularte E, Sikora G. Colônia Artificial De Abelhas – Um Algoritmo De Otimização Baseado Em Enxame De Abelhas E Uma Aplicação No Treinamento De Redes Neurais Para O Problema XOR. Universidade Estadual do Ceará, Brasil; 2012.

[12] Lopes H, Takahashi T. Computação Evolucionária. Curitiba: Omipax Editora Ltda, 2011.

[13] Weller T. Proposta de uma abordagem de busca tabu para o layout de células de manufatura. Universidade Tecnológica Federal do Paraná; 2008.

[14] Souza C, Mantovani J. Planejamento e projeto de circuitos secundários de distribuição de energia elétrica por meio de algoritmo busca tabu. Universidade Estadual Paulista, 2006.

[15] Crama Y, Schyns M. Simulated Annealing for Complex Portfolio Selection Problems. European Journal of Operational Research 150 (2003) 546–571

[16] Kresta A, Slova K. Solving cardinality constrained portfolio optimization problem by binary particle swarm optimization algorithm. Department of Mathematical Methods in Economics, Faculty of Economics, VŠB-Technical University of Ostrava, Sokolská třída 33, 701 21 Ostrava, Czech Republic, 2012.

[17] Briza A, Naval P. Stock Trading System Based on the Multi-objective Particle Swarm Optimization of Technical Indicators on End-of-day Market Data. University of the Philippines-Diliman, Philippines (2009).

[18] Gonçalves C, Pamplona E, Montevechi J. Seleção de carteiras através do modelo de markowitz para pequenos investidores (com o uso de planilhas eletrônicas). Escola Federal de Engenharia de Itajubá, São Paulo, Brasil 2002.

[19] Fulga, C, Dedu S. A New Approach in Multi-Objective Portfolio Optimization Using Value-at-Risk Based Risk Measure. Information and Financial Engineering (ICIFE), 2nd IEEE International Conference; 2010.

[20] Jinchuan K, and Zhang C. Study on the Optimization of Portfolio Based on Entropy Theory and Mean-variance Model. Beijing Jiaotong University, China.2008.

[21] Markowitz H. Portfolio selection: efficient diversification of investments. New York: Wiley, 1959.

[22] Borchers B, Mitchell JE. An improved branch and bound algorithm for mixed integer nonlinear programs. Computers & Operations Research 1994; 21:359-67.

[23] Mills TC. Stylized facts on the temporal and distributional properties of daily FT-SE returns. Applied Financial Economics 1997; 7:599-604.

[24] Alberto Fernandez, Sergio Gomez. Portfolio selection using neural networks. Computers and Operations Research, vol. 34, no. 4, pp. 1177-1191, 2007.

[25] Baykasoglu A, Özbakır L, Tapkan P. Artificial Bee Colony Algorithm and Its Application to Generalized Assignment Problem. Swarm Intelligence: Focus on Ant and Particle Swarm Optimization, Book edited by: Felix T. S. Chan and Manoj Kumar Tiwari, ISBN 978-3-902613-09-7, pp. 532, December 2007, Itech Education and Publishing, Vienna, Austria

[26] Aghazadeh F, Meybodi MR. Learning Bees Algorithm For Optimization. International Conference on Information and Intelligent Computing IPCSIT vol.18 (2011), Singapore.

[27] Pham DT, Ghanbarzadeh A, Koç E, Otri S, Rahim S, Zaidi M. The Bees Algorithm – A Novel Tool for Complex Optimisation Problems. Manufacturing Engineering Centre, Cardiff University, 2006.

[28] Andrade L, Cunha C. Algoritmo de colônia artificial de abelhas para um problema de clusterização capacitado. Escola Politécnica da USP, Brasil (2011).

[29] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization, vol. 39, no. 3, pp. 459-471, 2007.

[30] Karaboga D, Basturk B. On the performance of artificial bee colony (ABC) algorithm, Applied Soft Computing, vol. 8, no. 1, pp. 687-697, 2008.

[31] Haqiqi K, Kazemi T. Ant Colony Optimization Approach to Portfolio Optimization. International Conference on Economics, Business and Marketing Management (2012).