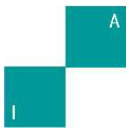
Inteligencia Artificial, 22(64) (2019), 85-101
doi: 10.4114/intartif.vol22iss64pp85-101



INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Uma Nova Meta-heurística Adaptativa Baseada em Vetor de Avaliações para Otimização de Portfólios de Investimentos

A New Adaptive Meta-Heuristic Based on a Vector Evaluated Approach for Portfolio Investiment Optimization

Letícia de Fátima Corrêa Costa[1], Omar Andres Carmona Cortes[2], João Pedro Augusto Costa[1]

[1]Programa de Pós-Graduação em Engenharia da Computação e Sistemas (PECS)
Universidade Estadual do Maranhão (UEMA)
São Luis, MA, Brasil
leticiadefsc@gmail.com

[2]Departamento de Computação (DComp)
Instituto Federal de Educação, Ciência e Tecnologia do Maranhão (IFMA)
São Luis, MA, Brasil
omar@ifma.edu.br

Abstract This article describes a new adaptive metaheuristic based on a vector evaluated approach for solving multiobjective problems. We called our proposed algorithm Vector Evaluated Meta-Heuristic. Its main idea is to evolve two populations independently, exchanging information between them, *i.e.*, the first population evolves according to the best individual of the second population and vice-versa. The choice of which algorithm will be executed on each generation is carried out stochastically among three evolutionary algorithms well known in the literature: PSO, DE, ABC. In order to evaluate the results, we used an established metric in multiobjective evolutionary algorithms called hypervolume. Tests have shown that the adaptive metaheuristic reaches the best hyper-volumes in three of ZDT benchmarks functions and, also, in two portfolios of a real-world problem called portfolio investment optimization. The results show that our algorithm improved the Pareto curve when compared to the hypervolumes of each heuristic separately.

Resumo Este artigo descreve uma nova meta-heurística adaptativa baseada em vetor de avaliações para solucionar problemas multiobjetivos. A ideia do algoritmo é evoluir duas populações independentes que trocam informações entre si, ou seja, a evolução de uma população se dá com base no melhor indivíduo da outra população e vice-versa, sendo que cada população pode ainda escolher, também de forma independente, qual meta-heurística (PSO, ED e ABC) utilizar durante sua execução. Para avaliar os resultados utiliza-se o hiper volume que é uma métrica comum em algoritmos evolutivos multiobjetivos. Testes demonstraram que a nova meta-heurística pode encontrar os melhores hiper volumes tanto nas funções ZDT de benchmarks quanto na otimização de portfólios de investimentos. Além disso, a nova meta-heurística adaptativa consegue gerar melhores fronteiras de Pareto do que meta-heurísticas de avaliação vetorial estáticas.

Keywords: Metaheuristics, multiobjective, vector evaluated, ABC, PSO, DE, portfolio optimization.

Palavras-Chave: Meta-heurísticas, multiobjetivo, avaliação vetorial, ABC, PSO, ED, otimização de portfólios.

ISSN: 1137-3601 (print), 1988-3064 (on-line)

1 Introdução

Usualmente, problemas do mundo real são naturalmente multiobjetivos, sendo formados por dois ou mais funções objetivo que conflitam entre si, ou seja, considerando duas funções quaisquer, a melhora em uma função implica na deterioração da outra. Para solucionar esse tipo de problema, métodos tradicionais, como por exemplo, os métodos baseados em gradiente, não são adequados [5], pois: *(i)* as funções precisam ser diferenciáveis, *(ii)* o espaço de busca é limitado a poucas variáveis; e *(iii)* os métodos clássicos tendem a não tratar as restrições. Nesse contexto, surgem os algoritmos evolutivos e de enxame (MOEA) como uma alternativa à resolução de problemas multiobjetivos (MOP), pois são nitidamente reconhecidos como adequados para esse tipo de problema, pois possuem a habilidade de encontrar várias soluções não dominadas em uma única execução [29].

A primeira aparição de um MOEA ocorreu em 1985 [23], quando David Shcaffer propôs o *Vector Evaluated Genetic Algorithm* (VEGA). O principal problema do VEGA é que ele tende a perder rapidamente a diversidade, pois pode haver a escolha de um super indivíduos nas operações genéticas. Dessa forma, não há uma boa distribuição das soluções sobre a curva de Pareto. Em 1993, Fonseca e Flemming [10] proporam um algoritmo multiobjetivo chamado *Multi-Objective Genetic Algorithm* (MOGA), que foi o primeiro a considerar a dominância entre as soluções. Desde então, diversos algoritmos, tais como, NSGA-II [6], SPEA2 [31], VEPSO [21], VEABC [19], VEDE [20], VEPBIL [3], dentre outros, têm sido propostos. Este trabalho tem um particular interesse nos algoritmos baseados em avaliação vetorial como o VEPSO, VEABC e VEDE. Como o VEPBIL foi proposto para problemas discretos o mesmo não é adequado para problemas contínuos. O interesse nos algoritmos VE se dá pelas seguintes razões:

i. Os algoritmos VE são adequados para problemas cuja quantidade de objetivos é par, sendo que muitos problemas do mundo real são baseados em dois objetivos.

ii. São fáceis de implementar se comparados a algoritmos MO tradicionais, pois cada função é tratada de forma simples por cada subpopulação.

iii. Por ser baseada em subpopulações, o algoritmo é facilmente paralelizável em diferentes modelos de programação paralela

Com base nas razões apresentadas, este trabalho propõe uma meta-heurística adaptativa baseada em VE para problemas multiobjetivos. Particularmente, três algoritmos foram implementados para formar a meta-heurística: evolução diferencial (DE) [25], otimização por nuvem de partículas (PSO) [15] e colônia de abelhas artificias (ABC) [13]. O objetivo é que cada subpopulação utilize uma dessas meta-heurísticas em um dos objetivos, sendo que a escolha de qual delas utilizar é feita de forma adaptativa estocástica, ou seja, as probabilidades de escolha de cada meta-heurística vão sendo alteradas à medida que melhores soluções vão sendo geradas. A ideia é que a melhor meta-heurística seja escolhida em tempo de execução gerando soluções de melhor qualidade sem a necessidade de ter que executar cada combinação de meta-heurística previamente para determinar qual a melhor delas. Além disso, a escolha em tempo de execução pode permitir solucionar uma gama maior de problemas do que as meta-heurísticas sem adaptação.

A adaptação estocástica tem sido usada com sucesso, como por exemplo, no trabalho de Ribeiro Jr. [2], no qual escolhe-se em tempo de execução quais operadores de cruzamento e quais de mutação são usados em um Algoritmo Genético (GA). Já em Borges et. al [1] recomenda-se estocasticamente qual meta-heurística utilizar dentre GA, PSO e DE em problemas com objetivo único. Já em aplicações multiobjetivo, Costa [4] escolhe estocasticamente entre GA, PSO e DE para fazer a atualização da população no SPEA2 em busca de novas soluções não dominadas. Nesse âmbito, este trabalho faz a adaptação escolhendo qual meta-heurística utilizar em cada população independente e formando a fronteira de Pareto de forma cooperativa. Os detalhes de seu funcionamento são dados nas próximas seções.

Nesse contexto, este artigo está dividido da seguinte maneira: a Seção 2 apresenta os conceitos básico de otimização em problemas multiobjetivos; a Seção 3 introduz os conceitos das meta-heurísticas usadas em cada subpopulação; a Seção 4 detalha o algoritmo proposto neste trabalho; a Seção 5 trata dos experimentos realizados e resultados encontrados; finalmente, a Seção 6 apresenta as conclusões deste trabalho e direcionamentos para futuros trabalhos.

2 Problemas Multiobjetivos

Um problema de otimização multiobjetivo (MOP) possui dois ou mais objetivos que devem ser otimizados, sendo que esses objetivos devem ser conflitantes, isto é, a melhora de um objetivo significa necessariamente a piora em outro. Dessa forma, um MOP com restrições de domínio tem a forma apresentada na Equação 1, na qual m é a quantidade de funções, i representa a quantidade de variáveis do vetor x, e L_i, L_s são respectivamente o limite inferior e o limite superior de cada variável.

$$\begin{aligned} Min \quad ou \quad & Max\ f_m(x), m = 1,2,\cdots,M \\ & L_i \leq x_i \leq L_s, 1,2,\cdots,D \end{aligned} \tag{1}$$

Diferentemente dos problemas de otimização com objetivo único, onde apenas o espaço de estado (ou espaço de busca) interessa, nos MOPs é necessário mapear o espaço de estados (representado por $\overrightarrow{x}$) para um espaço de objetivos representado por $f_m(x)$ como mostra a Figura 1, que apresenta o mapeamento de 7 (sete) soluções no espaço de estados para o espaço de funções, considerando uma dimensão tridimensional para $\overrightarrow{x}$ e duas funções objetivo.

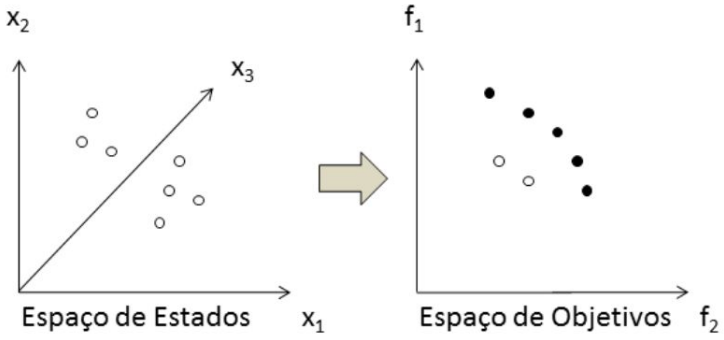


Figura 1: Mapeamento entre espaço de estados e espaço de objetivos. Adaptado de [5].

Além do mapeamento, a Figura 1 mostra um conceito importante na resolução de MOPs usado em diversos algoritmos evolutivos e de enxame, que é a dominância. No espaço de objetivos as 5 soluções em preto representam as soluções não dominadas, enquanto que as mais claras são soluções dominadas. O conjunto de soluções não dominadas é um grupo de soluções na qual nenhuma solução é melhor do que a outra, formando o que se chama de *fronteira de Pareto*. Assim, uma solução x_k domina uma solução x_i se: *(i)* x_k não é pior do que x_i em todas as funções objetivo; e, *(ii)* a solução x_k é estritamente melhor do que x_i em pelo menos um objetivo. Matematicamente diz-se que:

$$x^k \preceq x^i$$

. As próximas seções apresentam os problemas usados neste trabalho: benchmarks e o problema de otimização de portfólios de investimentos.

2.1 Benchmarks ZDT

As funções de *benchmark*, conhecidas por ZDTs, estão formalmente apresentadas na Tabela 1. As ZDTs são funções de minimização específicas utilizadas para avaliar a qualidade das soluções obtidas por algoritmos multiobjetivos [9]. Na Tabela 1 estão também as configurações para as funções de avaliação f_1 e f_2, o domínio do problema e a quantidade de genes que devem ser utilizadas para execução de cada teste. Para avaliar o resultado utiliza-se o conjunto de soluções obtidas nos testes e compara-se com a fronteira de Pareto. Assim, quanto mais próximo o conjunto de soluções estiver da fronteira de Pareto real, melhor é o conjunto de soluções.

Tabela 1: Funções ZDT's

Nome	Funções	Domínio
ZDT1	$\vec{f}: \Re^{30} \longrightarrow \Re^2$, onde	$x_i \in [0,1]$
	$f_1 = X_1$	$i = 1,\ldots,30$
	$g(x) = 1 + (9/n-1) * (\sum_{i=2}^{k} x_i)$	
	$f_2 = g(x) * [1 - \sqrt{x_1/g(x)}]$	
ZDT2	$\vec{f}: \Re^{30} \longrightarrow \Re^2$, onde	$x_i \in [0,1]$
	$f_1 = X_1$	$i = 1,\ldots,30$
	$g(x) = 1 + (9/n-1) * (\sum_{i=2}^{k} x_i)$	
	$f_2 = g(x) * [1 - (x_1/g(x))^2]$	
ZDT3	$\vec{f}: \Re^{30} \longrightarrow \Re^2$, onde	$x_i \in [0,1]$
	$f_1 = X_1$	$i = 1,\ldots,30$
	$g(x) = 1 + (9/n-1) * (\sum_{i=2}^{k} x_i)$	
	$f_2 = g(x) * [1 - \sqrt{x_1/g(x)} - x_1/g(x) * \sin(10\pi x_1)]$	
ZDT4	$\vec{f}: \Re^{10} \longrightarrow \Re^2$, onde	$x_1 \in [0,1]$,
	$f_1 = X_1$	$x_i \in [-5,5], \forall i \neq 1$
	$g(x) = 1 + 10 * (n-1) + (\sum_{i=2}^{k} ({x_i}^2 - 10 * \cos(4\pi x_i))$	$i = 1,\ldots,10$
	$f_2 = g(x) * [1 - \sqrt{x_1/g(x)}]$	
ZDT6	$\vec{f}: \Re^{10} \longrightarrow \Re^2$, onde	$x_i \in [0,1]$
	$f_1 = 1 - e^{-4x_1} * \sin^6(6\pi x_1)$	$i = 1,\ldots,10$
	$g(x) = 1 + 9 * [\sum_{i=2}^{n} x_i/(n-1)]^{0.25}$	
	$f_2 = g(x) * [1 - (f_1/g(x))^2]$	

2.2 O Problema da Otimização de Portfólios

O modelo clássico de Markowitz [16] [1] para o problema de seleção de portfólio assume que os investidores preferem maximizar o retorno dentro de um certo nível de risco ou minimizar o risco dentro de um certo nível de retorno. Este modelo é conhecido como modelo média-variância por utilizar estas estatísticas sobre o histórico dos preços normalizados para calcular, respectivamente, o retorno e o risco esperados para o portfólio [26].

O risco é avaliado como a variância do retorno do portfólio e depende não só em como muitos ativos individuais variam o retorno, mas também como eles variam em relação aos outros. Sendo assim, a matriz de covariância da junção da distribuição dos retornos é um adicional para o valor esperado [27].

Portanto, um portfólio financeiro otimizará dois objetivos conflitantes, ou seja, a maximização do retorno do portfólio esperado e a minimização da variância do retorno do portfólio [27] (minimização do risco). Formalmente é descrito como mostrado na Equação 2, na qual N é o número de ativos no portfólio, isto é, a dimensionalidade do problema de otimização; w_i é o peso do i-ésimo ativo a ser otimizado; σ^2 é o desvio do risco do portfólio, enquanto σ_{ij} é a covariância entre o ativo i e o ativo j. Se i for igual a j, σ_{ij} representa a variância de um particular ativo, r_p é a média de retorno do portfólio e r_i é a média de determinado ativo i [26].

$$\begin{aligned} \text{Min } \sigma^2 &= \sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j \sigma_{ij} \\ \text{Max } r_p &= \sum_{i=1}^{N} w_i r_i \\ \text{Sujeito a } \sum_{i=1}^{N} w_i = 1,\ w_i \geq 0,\ i = 1,\ldots,N \end{aligned} \tag{2}$$

[1] Markowitz foi reconhecido em 1990, com o Prêmio Nobel de Ciências Econômicas pela sua contribuição na teoria de seleção de portfólios. [17]

A fronteira eficiente de Markowitz correspondem aos portfólios que apresentam o maior retorno esperado para determinado nível de risco. A Figura 2 apresenta a fronteira eficiente. Nesta figura os portfólios 'A', 'B' e 'C' são considerados eficientes. Já o portfólio 'D' é considerado ineficiente, pois o portfólio B apresenta o mesmo risco, mas com maior retorno. Para este exemplo, um investidor não deverá escolher a carteira 'D', pois existe uma segunda alternativa com uma relação risco-retorno mais atrativa, ou seja, para um mesmo nível de risco da carteira 'D' existe a carteira 'B' que apresenta o melhor retorno esperado. Resumindo, os investidores devem considerar que um dos princípios para avaliar um investimento é o tipo de risco associado e o retorno que pode oferecer. Nota-se também que a fronteira de Pareto é um subconjunto da fronteira de Markowitz, pois na fronteira de Pareto D é considerado como um ponto dominado pelos demais portfólios.

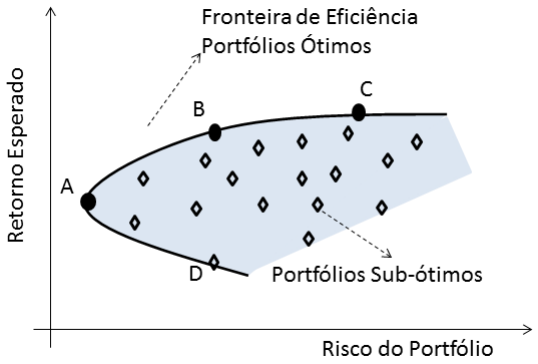


Figura 2: Fronteira Eficiente de Markowitz.

3 Meta-heurísticas

Meta-heurísticas são algoritmos que coordenam procedimentos de busca global e local com estratégias para escapar de mínimos locais em espaços de busca com soluções complexas [11], devendo ser utilizadas em problemas sobre os quais se tem pouca informação e/ou é muito custoso computacionalmente utilizar algoritmos enumerativos, ou seja, não é possível testar todas as possibilidades. A seguir são detalhadas as meta-heurísticas utilizadas neste trabalho.

3.1 Otimização por Nuvem de Partículas

Os pesquisadores Kennedy e Eberhart [15] desenvolveram a otimização por nuvem de partículas, em inglês chamada de *Particle Swarm Optimization* (PSO) [15]. O algoritmo PSO foi criado segundo a metodologia de sistemas sociais baseando-se no comportamento coletivo de indivíduos que interagem entre si e com o ambiente em torno de um ponto contendo comida ou local para descanso. Mais precisamente, é um algoritmo de enxame baseado no comportamento de como os pássaros buscam alimentos. Este tipo de sistema é conhecido também como inteligência de enxame. O Algoritmo 1 apresenta o PSO em pseudocódigo.

A inicialização das partículas (X_i) se dá de forma aleatória dentro do domínio das dimensões do problema. A velocidade das partículas (V_i) pode ser tanto inicializada por um valor aleatório quanto por um valor fixo. Após calcular o *fitness* de todo o enxame, atualiza-se a posição global (g), que é a posição da melhor solução encontrada (o *fitness* de g é chamado de *gbest*), e o histórico de cada partícula (P), que na inicialização vai conter os mesmos valores de inicialização e de avaliação do enxame. Ao entrar no laço a velocidade das partículas é atualizada a partir da equação 3, na qual w é uma constante de inércia, c_1 e c_2 são constantes aceleradoras, r_1 e r_2 são números aleatórios no intervalo $[0,1]$, g é a posição da melhor partícula (melhor solução) e p_k é o melhor valor encontrado pela partícula k.

$$v_{k+1} = wv_k + c_1 * r_1(p_k - x_k) + c_2 * r_2(g - x_k) \tag{3}$$

Em seguida, a posição da partícula é atualizada pela equação 4, ou seja, a nova posição é computada pela posição atual mais a velocidade recém calculada.

Algorithm 1 - Particle Swarm Otimization

Require: Inicializar partículas e velocidades;
1: Avaliar o enxame;
2: Atualizar global e histórico;
3: **repeat**
4: Calcular a velocidade das partículas;
5: Atualizar as posições das partículas;
6: **if** posição atual melhor que histórico **then**
7: Atualizar histórico;
8: **end if**
9: **if** posição atual melhor que global **then**
10: Atualizar global;
11: **end if**
12: **until** o critério de parada ser atingido.

$$x_{k+1} = x_k + v_{k+1} \tag{4}$$

Se x_{k+1} ultrapassar o domínio, o mesmo pode ser redefinido como o limite máximo ou mínimo que foi ultrapassado. Deve-se observar também que a velocidade pode também ser limitada no intervalo $[v_{min}, v_{max}]$. Essa restrição evita que haja uma explosão de velocidade nas partículas, o que geraria problemas tanto de exploração quanto na busca local do algoritmo.

3.2 Colônia Artificial de Abelhas

O algoritmo *Artifical Bee Colony* (ABC) foi proposto por Karaboga e Basturk [13], sendo baseado no comportamento das abelhas na busca por alimentos. No seu comportamento real, o principal objetivo das abelhas é aumentar o estoque de alimentos na colmeia, por isso as abelhas estão sempre em busca de fontes de alimentos para serem exploradas. Nesse contexto, existem três possibilidades de ações para as abelhas: *(i)* buscar novas fontes de alimento; *(ii)* explorar fontes já conhecidas; e, *(iii)* abandonar fontes de alimentação esgotadas. Como consequência das atividades, existem três tipos de abelhas: operárias, observadoras e exploradoras.

No algoritmo as abelhas operárias têm as coordenadas de uma fonte de alimento, ou seja, cada operária é responsável por uma solução. As operárias por sua vez compartilham suas informações com as abelhas observadoras, que tem por objetivo buscar novas fontes de alimento em regiões próximas às fontes atuais. As abelhas cuja fonte de alimento já se esgotou são transformadas em exploradoras e iniciam sua busca por novas fontes de alimento de forma aleatória [14]. Em termos computacionais, uma fonte de alimento representa uma solução possível para o problema, sendo que a qualidade do néctar que ali se encontra corresponde à qualidade da solução, ou seja, corresponde ao seu *fitness* ou aptidão. O algoritmo ABC executa os passos apresentados no Algoritmo 2:

Algorithm 2 - Artificial Bee Colony.

Require: Inicializar parâmetros da população;
1: **repeat**
2: Posicione a operária em uma fonte de alimento;
3: Posicione as observadoras em uma fonte de alimento;
4: Envie as exploradoras na busca por novas fontes;
5: Atualizar a melhor fonte de comida encontrada até o momento;
6: **until** alcançar o critério de parada.

Em cada ciclo ou iteração, a busca consiste em três passos: enviar a operária a uma fonte de alimento e medir a qualidade do néctar; selecionar quais fontes serão utilizadas pelas observadoras e determinar a quantidade de néctar; determinar quais abelhas serão transformadas em exploradoras e enviá-las para novas possíveis fontes. Quando o

algoritmo inicia, determina-se para as operárias um conjunto aleatório de fontes de alimento e quanto néctar cada um possui. Esse processo equivale a enviar as operárias para a fonte de alimento, que quando retornam compartilham suas informações com as observadoras. As operárias retornam para a fonte de alimento conhecida e visitam novas fontes de alimento tomando como base sua informação visual, ou seja, na vizinhança da solução. As observadoras irão escolher a sua região com base na qualidade do néctar compartilhado pelas operárias, isto é, quanto maior a quantidade de néctar, maior será a possibilidade da abelha observadora escolher essa fonte. Chegando a fonte estas também escolhem novas fontes de alimentação com base em sua vizinhança. Quando o néctar de uma fonte de alimento é abandonado, uma abelha exploradora escolhe uma nova fonte de alimento de forma aleatória. No modelo proposto [13] e [14], no máximo uma abelha exploradora é enviada por ciclo. Além disso, a quantidade de abelhas operárias e de observadores é o mesmo.

Com relação ao alcance visual das abelhas, uma solução é originalmente alterada utilizando-se a Equação 5, na qual k é um número aleatório entre 1 e a quantidade de operárias, j é um número aleatório entre 1 e a respectiva dimensão do problema, sendo j necessariamente diferente de i, e ϕ_{ij} é um número aleatório no intervalo $[-1,1]$.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{5}$$

Zhu e Kwong [30], inspirados pelo PSO, proporam uma melhoria como mostrado na equação 6, na qual o novo termo é chamado de *gbest*; y_i é a melhor solução do enxame e φ é um número aleatório no domínio $[0,C]$, sendo C uma constante não negativa. Independentemente da equação sendo utilizada, se a qualidade do novo néctar é melhor que o anterior, então a fonte anterior é abandonada.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \varphi_{ij}(y_i - x_{ij}) \tag{6}$$

As abelhas observadoras escolhem quais fontes visitar por meio da probabilidade fornecida pela Equação 7, na qual f representa o *fitness* de cada abelha e S é a quantidade de fontes de alimento (que é igual a quantidade de abelhas operárias). Após escolhida a fonte de alimento, a abelha observadora produz uma modificação na solução, e se a qualidade do néctar da nova solução for melhor, a observadora abandona a fonte anterior.

$$p_i = \frac{f_i}{\sum_{n=1}^{S} f_n} \tag{7}$$

Quando uma modificação é feita, seja por uma operária ou por uma observadora, se v_{ij} ultrapassar o limite superior ou inferior da respectiva dimensão, este pode ser redefinido com o valor mínimo ou máximo, dependendo do limite que foi ultrapassado.

Outro ponto a ser observado no algoritmo ABC é que qualquer fonte de alimentação que não for melhorada, isto é, que não receber melhorias em uma determinada quantidade de ciclos, deve ser imediatamente abandonada. A melhoria na qualidade do néctar é considerada positiva se o *fitness* da nova solução for melhor ou igual ao anterior, assim permite-se que novas soluções sejam exploradas mesmo tendo a mesma qualidade de néctar da anterior.

3.3 Evolução Diferencial

A Evolução Diferencial, do inglês *Differential Evolution* (DE), é um algoritmo evolutivo criado por Storn e Price [25]. Seu funcionamento é muito semelhante ao de um algoritmo genético (GA) [18], porém enquanto os GA's executam cruzamento e mutação, na DE executam-se os operadores na ordem inversa, isto é, primeiro a mutação e depois o cruzamento. No Algoritmo 3 apresenta-se os passos executados pelo DE.

Assim como no PSO, no DE a população é inicializada de forma aleatória dentro do domínio de cada gene e seu *fitness* é avaliado. Ao entrar no laço cria-se o vetor de diferenças (v) de acordo com a Equação 8, na qual x_α, x_β *e* x_γ são três indivíduos escolhidos aleatoriamente dentro da população em que $\alpha \neq \beta \neq \gamma$, isto é, α, β *e* γ são diferentes, e (F) é um fator de mutação. Como os três indivíduos são escolhidos aleatoriamente, essa estratégia é chamada de *DE/Rand/Bin*. Quando x_α é substituído pelo melhor indivíduo da população, a estratégia passa a ser chamada de *DE/Best/Bin*.

$$v = x_\alpha + F(x_\beta - x_\gamma) \tag{8}$$

Algorithm 3 - Evolução Diferencial

Require: Inicializar parâmetros da população;
1: Avaliar população;
2: **for** de i = 1 até o tamanho da população **do**
3: Criar vetor de diferenças;
4: Formar novo indivíduo
5: **if** $fit(novo) < fit(individuo_i)$ **then**
6: Substituir novo no alvo
7: **else**
8: Manter individuo alvo
9: **end if**
10: **end for**

Após gerar o vetor de diferenças, o novo indivíduo será criado da seguinte forma: para cada gene do indivíduo sorteia-se um valor r_i no intervalo $[0,1]$, sendo que i corresponde a um gene que deve obedecer a restrição de seu domínio. Se $r_i < CR$, ou seja, se o número sorteado for menor que a taxa de cruzamento *(Crossover Rate - CR)* então o gene do novo indivíduo será originário de v, caso contrário, o gene virá do indivíduo alvo i. Essa operação é semelhante ao cruzamento discreto dos AGs e pode ser representado como mostrado na Equação 9.

$$\text{novo}_i = \begin{cases} v_i, & \text{se} \quad r \leq CR \\ pop_i & \text{se} \quad r > CR \end{cases} \tag{9}$$

Após o cruzamento deve-se verificar se o novo indivíduo apresenta um *fitness* melhor que indivíduo alvo, em caso afirmativo o novo indivíduo substitui o indivíduo alvo, caso contrário o alvo permanece na população.

4 Meta-Heurística Adaptativa - AVEMH

Como mencionado anteriormente, as abordagens vetoriais são fáceis de implementar para problemas cuja quantidade de objetivos é par; e como evoluem a partir de subpopulações independentes, essas abordagens são fortes candidatas à paralelização. A Figura 3 mostra a ideia de como os algoritmos VE funcionam para dois objetivos, no qual cada subpopulação trabalha com um único objetivo. A chave que os faz funcionar é que a atualização de cada subpopulação é feita com base no melhor indivíduo da outra população.

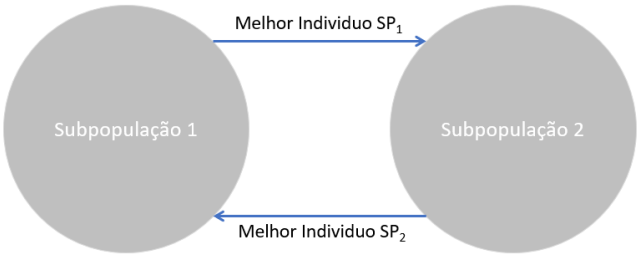


Figura 3: Mapeamento entre espaço de estados e espaço de objetivos. Adaptado de [5].

Matematicamente, considerando duas populações independentes P_1 e P_2 e sendo seus respectivos melhores indivíduos x_1 e x_2, então x_1 guia a atualização de SP_2 e x_2 guia a atualização de SP_1. É importante notar que não é necessário determinar a dominância neste ponto, pois as atualizações na posições de x são feitas pelos indivíduos das subpopulações opostas. Dessa forma, direciona-se SP_1 na direção de x_2 e SP_2 na direção de x_1. Esse comportamento tende a distribuir as soluções pela fronteira de Pareto.

Assim, quando se utiliza o *Vector Evaluated PSO* (VEPSO), por exemplo, a atualização da velocidade (v) na SP_1 usará o melhor indivíduo (g) da SP_2 e vice-versa. Com relação à DE, por estar baseado no melhor indivíduo, deve-se utilizar a estratégia $DE/Best/1$, caso contrário, uma forma de estabelecer dominância entre os indivíduos trocados

deve ser estabelecida, o que acaba saindo um pouco da filosofia da avaliação vetorial. Já com relação ao algoritmo ABC, a troca entre as subpopulações é feita conforme o melhor néctar. Essa estratégia faz com que as soluções não se acumulem nos extremos da fronteira de Pareto. O fluxograma do AVEMH é apresentado na Figura 4.

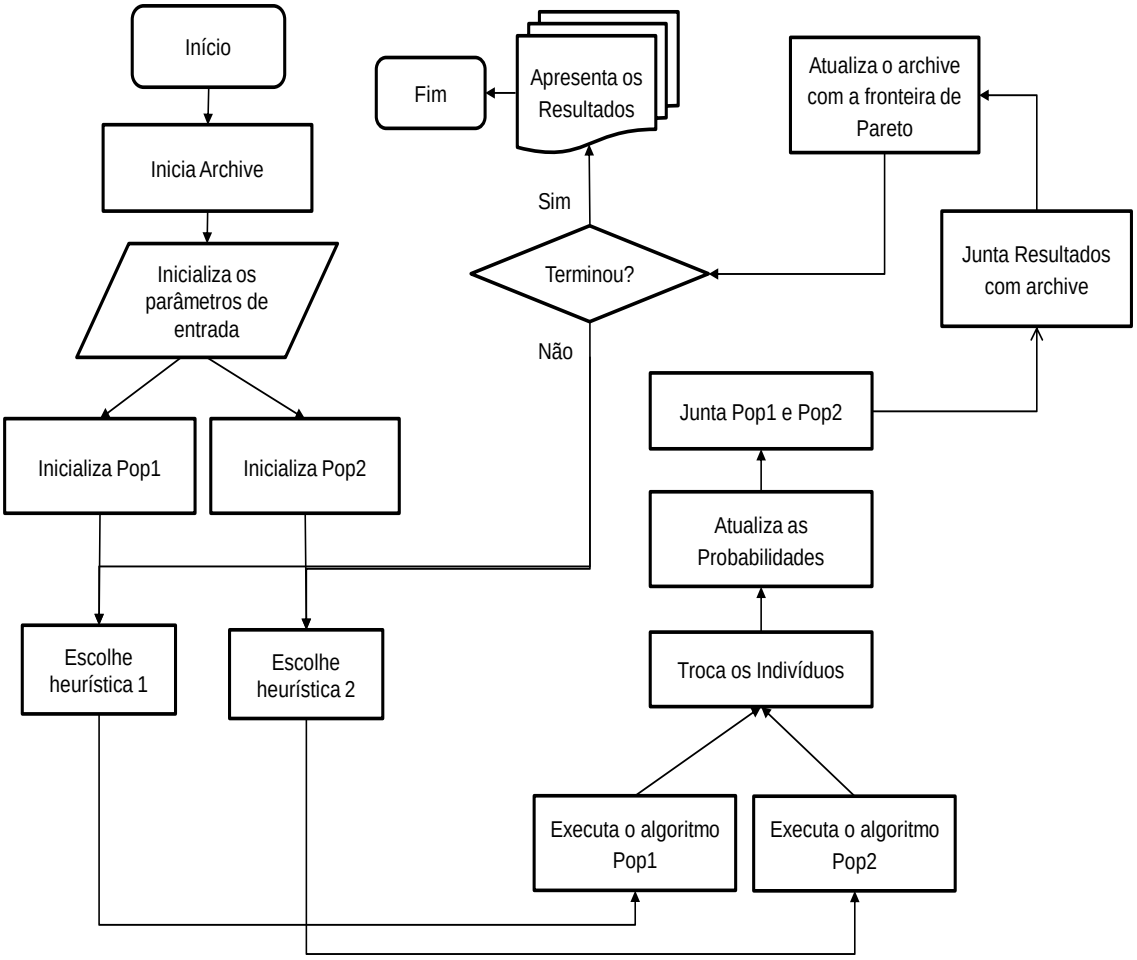


Figura 4: Fluxograma AVEMH

No começo, o AVEMH inicializa as probabilidades de cada heurística a ser escolhida de forma uniforme. Como são três possibilidades (PSO, DE e ABC) cada uma é inicializada com 0.33. Após serem definidas as probabilidades, as subpopulações são inicializadas de forma aleatória dentro do domínio do problema. Dentro do laço de iterações ocorre a escolha de qual meta-heurística utilizar, ocorre a troca de indivíduos entre subpopulações para que ocorra a atualização das soluções no espaço de busca. Em seguida, em cada subpopulação verifica-se se houve melhora no *fitness* da função.

A avaliação da população pode ser feita com duas abordagens. A primeira considera que se o melhor indivíduo veio da outra população então assume-se que houve uma melhora no ótimo da função. A outra abordagem utiliza a média da qualidade dos indivíduos da população, caso esta tenha melhorado com relação à média anterior, entende-se também que houve uma melhora na população.

Se houve melhora, então a heurística que enviou seu melhor indivíduo tem sua probabilidade aumentada, caso contrário, é diminuída. Nesse momento podem acontecer dois casos: *(i)* se houve uma melhora, a probabilidade da heurística que enviou o indivíduo é incrementada em 0.01, enquanto que as demais são diminuídas em 0.005; e *(ii)* se não houve melhora, a heurística que enviou o indivíduo tem sua probabilidade decrementada em 0.01 e as demais tem sua probabilidade aumentada em 0.005. Finalmente, as populações são unidas e as funções adjacentes

são calculadas, ou seja, calcula-se f_2 em Pop_1 e f_1 em Pop_2. Em seguida os resultados são unidos ao *archive* de modo a determinar a nova fronteira de Pareto, isto é, as soluções dominadas são removidas do *archive*. Dado que as populações independentes não são alteradas e o laço re-inicia, não é necessário que haja uma etapa de separação.

Este modelo generalizado deve ser robusto o suficiente para conseguir resolver diversos problemas multiobjetivo. Para tanto, a meta-heurística AVEMH pode ser aplicada tanto em funções de *benchmark* quanto na resolução de problemas do mundo real, tais como, *Despacho Econômico e Ambiental de Energia* e a *Seleção de Portfólios*, dentre outros. O algoritmo 4 mostra a sequencia de operações em pseudocódigo de modo a facilitar seu entendimento.

Algorithm 4 - Algoritmo Multiobjetivo Vetorial - AVEMH

Require: Iniciar archive vazio
1: Iniciar Probabilidades de Pop_1
2: Iniciar Probabilidades da Pop_2
3: Escolher heurística da Pop_1
4: Escolher heurística da Pop_2
5: Inicializar parâmetros da população
6: **repeat**
7: Trocar indivíduos
8: Atualizar a subpopulação usando a meta-heurística escolhida
9: **if** Melhorar solução é verdadeiro **then**
10: Atualizar Probabilidades de quem enviou
11: **end if**
12: Calcular o *fitnesss* da função adjacente
13: Juntar as subpopulações no *archive*
14: Remover soluções dominadas do *archive*
15: **until** Alcançar o critério de parada

5 Experimentos

5.1 Termos, Configurações e Métricas

Aqui são apresentados os termos utilizados para nomenclatura, as configurações relacionadas ao ambiente de teste, as configurações de parâmetros utilizados para as execuções do AVEMH e a métrica adotada para avaliação dos resultados. Para uniformizar a nomenclatura, o melhor indivíduo será sempre chamado de *pBest*. No VEPSO o indivíduo migrado é o g. No VEDE o indivíduo migrado é o que apresenta melhor *fitness* na população. Já no VEABC será migrada a localização do melhor néctar. Todos os testes foram realizados em um computador com processador Intel(R) Core(TM) i5 com 2.50GHz, 8,00 GB de RAM e com sistema operacional Windows 8.1. de 64 bits. A implementação foi feita em linguagem Java versão 8 e NetBeans 8.2. As configurações das meta-heurísticas estão apresentadas na Tabela 2. Cada experimento foi executado 31 vezes para que sua distribuição seja considerada normal de acordo com o teorema do limite central, sendo possível assim a utilização de testes paramétricos na comparação dos resultados. Os parâmetros foram definidos como em [28], [3].

Tabela 2: Configurações usadas por cada meta-heurística

Meta-heurística	Parâmetro	Valor
VEPSO	inercia	1/(1/2 + ln(2))
	r_1, r_2	1/2 + ln(2)
VEABC	sobrevivem	90%
VEDE	taxa de crossover	0.9
	fator de mutação	0.7

A métrica utilizada nos testes é o hiper volume apresentado na Equação 10, na qual v_i representa um hipercubo construído a partir das soluções da fronteira de Pareto (conjunto Q). Assim, o hiper volume total é calculado pela

união de todos os hipercubos v_i. Em outras palavras, o hiper volume é a área que esta sob a fronteira de Pareto. Assim, um maior o hiper volume pode tender a apresentar melhores fronteira de Pareto.

$$h_v = volume(\bigcup_{i=1}^{|q|} v_i) \tag{10}$$

5.2 Resultados nas Funções de Benchmark

Na Tabela 3 estão os valores calculados referentes ao maior, a média, a variância e o desvio padrão dos hiper volumes para as ZDTs, na qual pode-se observar que o valor real e o encontrado pela AVEMH são próximos. Na Tabela 4 é apresentada a quantidade de soluções encontradas pelo AVEMH em relação aos melhores hiper volumes encontrados para as ZDT's. Já a Figura 5 apresenta como as soluções encontradas se posicionam em relação à fronteira de Pareto Real.

Tabela 3: Resultados Calculados pelo AVEMH para as ZDTs

Função	Real Hv	Melhor Hv	Média Hv	Variância	D. Padrão
ZDT1	120.6621	120.6616	120.5367	0,000238	0.015440
ZDT2	120.3288	120.3245	120.2548	0,000138	0.011745
ZDT3	128.7781	128.7714	128.7714	3E-26	1E-13
ZDT4	120.6661	120.6621	120.6504	2E-26	1E-13
ZDT6	117.5149	117.4579	114.4111	0.27E+3	16.4851

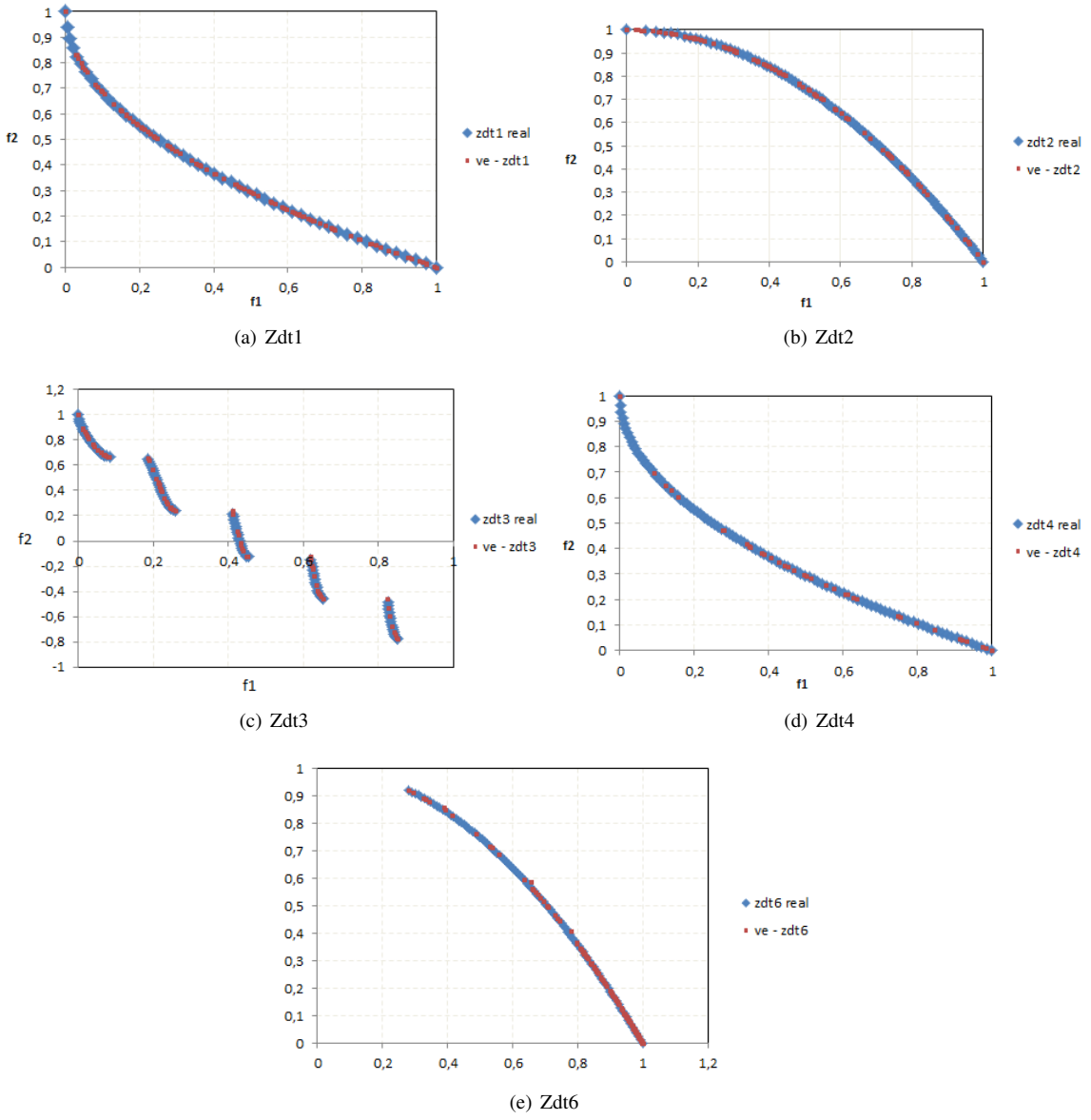
Tabela 4: Hiper volumes atingidos pelo AVEMH para as ZDT's.

	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
Nº de gerações			100		
Dimensão	30	30	30	10	10
Nº de soluções na fronteira	200	73	33	59	10
Hiper volume - Melhor	120,6616	120,3245	128,7714	120,6621	117,4579
Hiper volume - Médio	120,6615	120,3230	128,7258	120,5771	117,1978

Ainda nas ZDTs, a Tabela 5 faz um comparativo dos hiper volumes entre a AVEMH e os algoritmos VE sem adaptação. O AVEMH alcança melhores resultados nas ZDT1, ZDT2 e ZDT3. O VEABC encontra o melhor hiper volume na ZDT4 e o VEDE o melhor resultado na ZDT5, muito embora todos os valores sejam muito parecidos. Com intuito de verificar se realmente existem diferenças significativas entre os resultados apresentados pelo AVEMH, VEPSO, VEABC e VEDE foi realizada uma análise de variância (ANOVA) sobre os dados obtidos nos hiper volumes com significância de 95% (α = 0,05) e um teste de Tukey apresentados nas Tabelas 6 e 7. Como se pode observar, o teste ANOVA indica que há diferenças significativas em todas as funções ZDTs.

Tabela 5: Comparativos dos melhores resultados atingidos pelo AVEMH em relação aos melhores resultados atingidos com VEPSO, VEABC e VEDE, nas funções ZDTs.

Função	Real Hv	AVEMH Hv	VEPSO Hv	VEDE Hv	VEABC Hv
ZDT1	120.6621	**120.6616**	120.6547	120.6161	120.6562
ZDT2	120.3288	**120.3245**	120.3174	120.2431	120.3244
ZDT3	128.7781	**128.7714**	128.5893	128.6462	128.6250
ZDT4	120.6661	120.6621	120.6633	120.6607	**120.6645**
ZDT6	117.5149	117.4579	117.2347	**117.5116**	117.1973



(a) Zdt1 (b) Zdt2

(c) Zdt3 (d) Zdt4

(e) Zdt6

Figura 5: Comparativo da fronteira real das funções de *benchmark* com as fronteiras encontradas pelo AVEMH para as ZDTs.

O teste de Tukey na Tabela 7 indica que o AVEMH e VEABC obtiveram o melhor desempenho na ZDT1 já que a diferença não é significativa. Nas ZDT2 e ZDT3, o AVEMH obteve realmente os melhores resultados. O algoritmo VEABC juntamente com o VEPSO obtiveram o melhor resultado na ZDT4. E na ZDT6, o algoritmo VEDE realmente obteve os melhores resultados.

5.3 Resultados na Otimização de portfólios

Para a resolução da Seleção de Portfólios fez-se um comparativo dos resultados encontrados pelo AVEMH com a fronteira eficiente dos arquivos disponíveis para testes em `http://people.brunel.ac.uk/~mastjjb/jeb/orlib/portinfo.html`. Neste trabalho, os conjuntos de dados são os 'Hong Kong Hang Seng', 'German Dax 100','British FTSE 100', 'US S&P 100' e 'Japonese Nikkei' que são compostos por 63, 85, 89, 98 e 225 ativos, respectivamente.

Tabela 6: Teste ANOVA comparando o AVEMH com as demais meta-heurísticas nas funções ZDT

F crítico = 2,73									
Função	F	Função	F	Função	F	Função	F	Função	F
ZDT1	11,78	ZDT2	653,6	ZDT3	3888	ZDT4	10,4	ZDT6	100,1

Tabela 7: Teste de Tukey comparando o AVEMH com as demais meta-heurísticas nas funções ZDTs

ZDT1					
	p		p		p
AVEMH vs VEABC	0,998	AVEMH vs VEDE	0,0007	AVEMH vs VEPSO	0,00006
	p		p		p
ZDT2					
	p		p		p
AVEMH vs VEABC	0,158	AVEMH vs VEDE	0,0000	AVEMH vs VEPSO	0,0000
ZDT3					
	p		p		p
AVEMH vs VEABC	0,0000	AVEMH vs VEDE	0,0000	AVEMH vs VEPSO	0,0000
ZDT4					
	p		p		p
AVEMH vs VEABC	0,0525	AVEMH vs VEDE	0,1738	AVEMH vs VEPSO	0,039
ZDT6					
	p		p		p
AVEMH vs VEABC	0,9986	AVEMH vs VEDE	0,0000	AVEMH vs VEPSO	0,0026

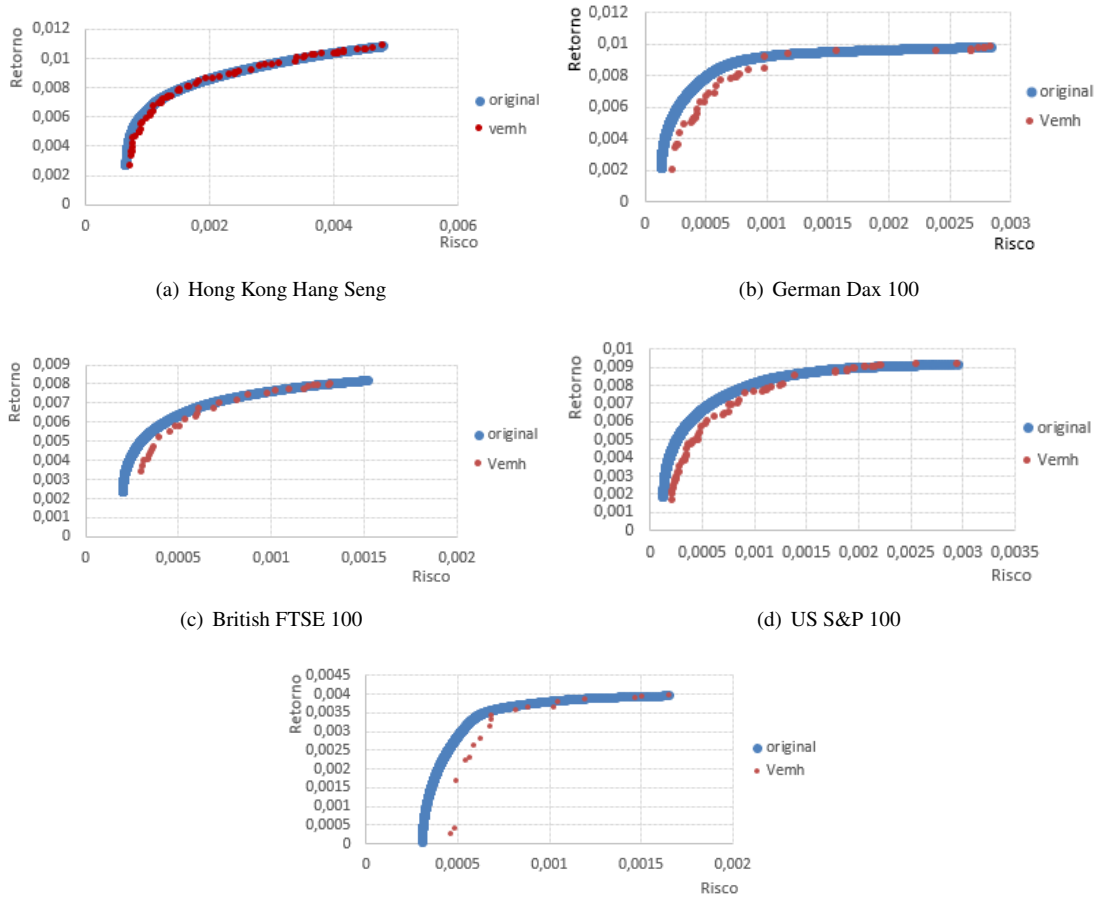
Nestes conjuntos de dados são disponibilizados a média dos retornos dos ativos, o desvio padrão dos retornos e a correlação entre esses ativos. O Hong Kong Hang Seng, conhecido oficialmente como 'Hang Seng Index' (HSI), é o principal índice da bolsa de valores de Hong Kong [22]. Já o índice German Dax, conhecido por 'DAX Índex', é formado pelas maiores empresa da bolsa de valores de Frankfurt na Alemanha [7]. O British FTSE é composto por 30 maiores empresas de capital aberto do Reino Unido, conhecido por 'FTSE' este índice afeta a economia britânica e mundial representando boa parte da capitalização na bolsa de valores de Londres [8]. O US S&P 100 é um índice do mercado de ações norte-americano mantido pela Standard & Poor's sendo formado por 101 empresas que possuem ações nas bolsas de valores NYSE e NASDAQ [12]. E por fim, o Japonese Nikkei, conhecido por 'Nikkei 225' é o principal índice econômico da bolsa de valores de Tóquio [24].

Na Tabela 8 são apresentados o número de gerações utilizadas, o número de ativos em cada problema, a quantidade de soluções encontradas e os hiper volumes calculados pelo AVEMH em cada uma das funções de testes quando da resolução da Seleção de Portfólios. A Figura 6 apresenta a comparação entre a fronteira de Pareto ótima e os testes realizados com o AVEMH e os conjuntos de dados do problema da Seleção de Portfólios.

Tabela 8: Hiper volumes atingidos no problema de seleção de portfólios

	HST	DAX	FTSE	S&P 100	Nikkei 225
Nº de gerações			2000		
Nº de ativos	31	85	89	98	225
Nº de soluções encontradas	65	58	36	180	30
Hiper volume - Melhor	120,9459	120,9724	120,9649	120,9650	120,9824
Hiper volume - Médio	120,8725	120,8901	120,9018	120,8815	120,8999

Na Tabela 9 estão disponíveis a média dos melhores resultados atingidos pelo AVEMH em relação aos resultados atingidos com o VEPSO, VEABC e VEDE, para os problemas da Seleção de Portfólios. Como se pode observar, o AVEMH apresenta uma média melhor nos portfólios HST, DAX e FTSE, enquanto que o VEDE apresenta uma melhor média no S&P 100 e Nikkei.



(a) Hong Kong Hang Seng

(b) German Dax 100

(c) British FTSE 100

(d) US S&P 100

(e) Japonese Nikkei

Figura 6: Comparativo da fronteira real das funções de *benchmark* com as fronteiras encontradas pelo AVEMH para a Seleção de Portfólios.

Com intuito de verificar se realmente existem diferenças significativas entre os resultados apresentados pelo AVEMH e as demais meta-heurísticas foi realizada uma análise de variância (ANOVA) com grau de significância de 95% ($\alpha = 0.05$) e F crítico igual a 2,73 conforme apresentado na Tabela 10. Como pode ser observado há diferença significativas nos portfólios HST e DAX. Para identificar as diferenças fez-se um teste de Tukey conforme apresentado na Tabela 11, na qual pode-se notar que as maiores diferenças estão a favor do AVEMH e em contra ao VEDE e a VEABC, ou seja, a AVEMH apresenta melhores resultados que o VEDE e o VEABC. Além disso, não há diferença significativa entre o AVEMH e o VEPSO.

A Tabela 12 apresenta o tempo médio de cada execução para cada portfólio em minutos. Em média o AVEMH é mais rápido que o VEABC na otimização do HST, S&Pe e Nikkei 225. E mais rápido que o VEDE em todos os portfólios. Embora o VEPSO seja o mais rápido, não é o algoritmo que consegue alcançar as melhores configurações de portfólios como previamente apresentado.

6 Conclusões e Trabalhos Futuros

Este trabalho apresentou uma meta-heurística adaptativa baseada em avaliação vetorial para resolução de problemas multiobjetivo, em especial aplicado na otimização de portfólios de investimentos. A principal vantagem da avaliação vetorial é ser de fácil implementação e não consumir tempo do algoritmo com operações baseadas em dominância. Além disso, a adaptação em tempo de execução, em teoria, permite solucionar uma gama maior de problemas sem a

Tabela 9: Resultados atingidos pelo AVEMH em relação aos resultados atingidos com VEPSO, VEABC e VEDE, para o problema de Seleção de Portfólio considerando a média do hiper volume.

Função	Gerações	AVEMH Hv	VEPSO Hv	VEABC	VEDE Hv
HST	2000	**120,8725**	120,8592	120,7398	120,8191
DAX	2000	**120,8901**	120,7752	120,8895	120,8898
FTSE	2000	**120,9018**	120,8869	120,8926	120,8979
S&P 100	2000	120,8815	120,8552	120,8620	**120,9067**
Nikkei 225	2000	120,8999	120,8666	120,8772	**120,9010**

Tabela 10: Teste ANOVA comparando o AVEMH com as demais metaheurísticas

F crítico = 2,73									
Portfolio	F	Portfolio	F	Portfolio	F	Portfolio	F	Portfolio	F
HST	5,596	DAX	5,603	FTSE	0,056	S&P 100	0,055	Nikkei	0,664

Tabela 11: Teste de Tukey comparando o AVEMH com as demais meta-heurísticas nos portfólios HST e DAX

HST					
	p		p		p
AVEMH vs VEABC	**0,0017**	AVEMH vs VEDE	**0.4432**	AVEMH vs VEPSO	0.9821
	p		p		p
DAX					
AVEMH vs VEABC	**0,0017**	AVEMH vs VEDE	**0.4424**	AVEMH vs VEPSO	0.9822

Tabela 12: Tempo de execução para a Seleção de Portfólios - 2000 iterações

	AVEMH		VEPSO		VABC		VEDE	
Dados	Média	Desvio	Média	Desvio	Média	Desvio	Média	Desvio
HST	0,30 m	0,0086 m	0,18 m	0,0075 m	0,34 m	0,0098 m	0,38 min	0,0416 m
DAX	1,03 m	0,0088 m	0,54 m	0,0200 m	0,90 m	0,0118 m	1,05 m	0,027 m
FTSE	1,00 m	0,0066 m	0,57 m	0,0183 m	0,98 m	0,0128 m	1,12 m	0,0338 m
S&P	1,04 m	0,009 m	0,71 m	0,0085 m	1,14 m	0,0056 m	1,26 m	0,0295 m
Nikkei 225	3,25 m	0,0268 m	2,47 m	0,0115 m	3,48 m	0,0278 m	3,72 m	0,1415 m

necessidade de explorar a combinação entre as meta-heurísticas utilizadas. O resultados mostraram que a AVEMH apresentou melhores hiper volumes nas funções de benchmark ZDT1, ZDT2 e ZDT3; e na otimização de portfólio de investimentos dos portfólios HST, DAX e FTSE. Por outro lado, algumas diferenças não foram estatisticamente significativas possivelmente pela quantidade de iterações utilizadas. Nesse contexto, espera-se em trabalhos futuros testar a meta-heurísticas em problemas mais complexos, como por exemplo o S&P 500, para verificar se a adaptação estocástica produz efetivamente melhores fronteiras de Pareto.

Outros trabalhos futuros incluem: (i) paralelização dos algoritmos tanto em arquiteturas multi-núcleo (*multicore*) quanto em arquiteturas com muitos núcleos (GPU - *many cores*); (ii) incluir a auto-adaptação no algoritmo na qual os parâmetros como F, w, c_1, c_2, dentre outros sejam incluídos nos genes dos indivíduos e decididos em tempo de execução; (iii) utilizar a lógica nebulosa para determinar qual meta-heurística utilizar; e (iv) aplicar o algoritmo AVEMH na otimização de parâmetros em algoritmos de aprendizagem de máquina.

É interessante ressaltar que futuramente esta meta-heurística pode ser paralelizável com o objetivo de melhorar o tempo de execução na resolução de problemas do mundo real mais custosos, tal como a seleção de portfólios usando o S&P 500. Também é viável o desenvolvimento de um modelo fuzzy para realização da análise de melhora da população. E ainda, tornar o algoritmo suficientemente robusto para a resolução de mais problemas do mundo real.

Referências

[1] H. P. Borges and D. Cortes, O. A. C.and Vieira. An adaptive metaheuristic for unconstrained multimodal numerical optimization. In P. Korošec, N. Melab, and El-G. Talbi, editors, *Bioinspired Optimization Methods and Their Applications*, pages 26–37, Cham, 2018. Springer International Publishing.

[2] E. Carvalho Jr., O. A. C. Cortes, J. P. Costa, and A. Rau-Chaplin. A stochastic adaptive genetic algorithm for solving unconstrained multimodal numerical problems. In *2016 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, pages 130–137, May 2016.

[3] O. A. C. Cortes, A. Rau-Chaplin, and P. F. do Prado. A new vector evaluated PBIL algorithm for reinsurance analytics. In *2015 Latin America Congress on Computational Intelligence (LA-CCI)*, pages 1–6, Oct 2015.

[4] J. P. A. Costa, E. Carvalho Jr., and O. A. C. Cortes. An adaptative algorithm for updating populations on SPEA2. In *XII Simpósio Brasileiro de Automação Inteligente - Porto Alegre - RS*, pages 78–83, 2017.

[5] K. Deb. *Multi-Objetive Optimization using Evolucionay Algorithm*. Jhon Wiley & Sons, LTD, 2001.

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Transaction on Evolutionary Computing*, 6(2):182–197, April 2002.

[7] Disford. Top 30 companies of germany in the dax index 2019. https://disfold.com/top-companies-germany-dax/, 2019.

[8] Disford. Top 30 companies of the uk in the ftse index 2019. https://disfold.com/top-companies-uk-ftse/, 2019.

[9] ETH. Systems optimization. https://sop.tik.ee.ethz.ch/download/supplementary/testpro-blems/, 2019.

[10] C. M. Fonseca and P. Fleming. Genetic algorithms for multiobjective optimization: Formulation discussion and generalization. *the fifth Intl conference on Genetic Algorithms*, 93, 02 1999.

[11] F. Glover and A. G. Kochenberger. *Handbook of MetaHeuristics*. Kluwer Academic Publishers Boston, New York, 2005.

[12] S&P Down Jones Indices. S&p 100. https://us.spindices.com/indices/equity/sp-100, 2019.

[13] D. Karaboga and B. Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of Global Optimization*, 39(3):459–471, Nov 2007.

[14] D. Karaboga and B. Basturk. On the performance of artificial bee colony (abc) algorithm. *Applied Soft Computing*, 8(1):687–697, 2008.

[15] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948, Nov 1995.

[16] H. Markowitz. Portfolio selection. *The Journal of Finance*, 7:77–91, March 1952.

[17] H. Markowitz, W. F. Sharpe, and M. H. Miller. The founders of modern finance: Their prize winning concepts and 1990 nobel lectures. *CFA Intitute*, 1990.

[18] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs (3rd Ed.)*. Springer-Verlag, Berlin, Heidelberg, 1996.

[19] S. N. Omkar, J. Senthilnath, R. Khandelwal, G. Narayana Naik, and S. Gopalakrishnan. Artificial bee colony (abc) for multi-objective design optimization of composite structures. *Applied Soft Computing*, 11(1):489 – 499, 2011.

[20] K. E. Parsopoulos, D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis. Vector evaluated differential evolution for multiobjective optimization. In *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, volume 1, pages 204–211, June 2004.

[21] K. E. Parsopoulos and M. N. Vrahatis. Particle swarm optimization method in multiobjective problems. In *Proceedings of the 2002 ACM Symposium on Applied Computing*, SAC '02, pages 603–607, New York, NY, USA, 2002. ACM.

[22] T. Reis. Hang seng: saiba mais sobre o principal índice da bolsa de Hong Kong. https://www.sunoresearch.com.br/artigos/hang-seng/, 2019.

[23] J Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the First Int. Conference on Genetic Algortihms, Ed. G.J.E Grefensette, J.J. Lawrence Erlbraum*, pages 93–100, 01 1985.

[24] Nikkei 225 Official Site. Nikkei stock average (nikkei 225). https://indexes.nikkei.co.jp/en/nkave/index/profile?idx=nk225, 2019.

[25] R. Storn and Kenneth Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, Dec 1997.

[26] B Y. Qu, Q Zhou, J M. Xiao, Jing Liang, and Ponnuthurai Suganthan. Large-scale portfolio optimization using multiobjective evolutionary algorithms and preselection methods. *Mathematical Problems in Engineering*, 2017:1–14, 02 2017.

[27] I. Yevseyeva, A. P. Guerreiro, M. T. M. Emmerich, and C. M. Fonseca. A portfolio optimization approach to selection in multiobjective evolutionary algorithms. In T. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith, editors, *Parallel Problem Solving from Nature – PPSN XIII*, pages 672–681, Cham, 2014. Springer International Publishing.

[28] M. Zambrano-Bigiarini, Maurice C., and R. Rojas. Standard particle swarm optimisation 2011 at cec-2013: A baseline for future pso improvements. In *IEEE Congress on Evolutionary Computation, CEC*, pages 2337–2344, 06 2013.

[29] W. Zhang and S. Fujimura. Improved vector evaluated genetic algorithm with archive for solving multiobjective pps problem. In *2010 International Conference on E-Product E-Service and E-Entertainment*, pages 1–4, Nov 2010.

[30] G. Zhu and S. Kwong. Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied Mathematics and Computation*, 217(7):3166–3173, 2010.

[31] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In *Proceedings of the EUROGEN*, volume 3242, 01 2001.