

Aplicação de Algoritmo Genético Multiobjetivo em Otimização de Portfólios

Simone Aparecida Miloca

Centro de Ciências Exatas, Universidade Estadual de Cascavel
Rua Universitária, 1619 - Cascavel, PR
smiloca@gmail.com

Thais Mariane Biembengut Faria

Programa de Pós-Graduação de Métodos Numéricos em Engenharia - UFPR*
thaismariane@gmail.com

Neida Maria P. Volpi

Departamento de Engenharia de Produção, UFPR*
neidavolpi@gmail.com

*Centro Politécnico, Jardim das Américas - Curitiba, PR

RESUMO

Neste artigo apresenta-se a aplicação de algoritmos genéticos multiobjetivo (MOEAs) no problema de otimização de portfólio. O objetivo do trabalho foi avaliar o desempenho dos algoritmos genéticos, NSGA-II e SPEA2, no problema de investimento em um conjunto de ações. Adotou-se para obtenção da fronteira eficiente, o modelo de média-variância proposto por Markowitz (1952). O conjunto de dados foi obtido de 26 ativos negociados no BOVESPA entre período 01 de janeiro de 2004 a 31 de dezembro de 2007. O desempenho dos algoritmos foi avaliado por meio das métricas C e S. O resultado obtido pela métrica C aponta que aproximadamente 61% das soluções obtidas pelo NSGA-II, são dominadas pelas soluções do SPEA. Com relação à métrica S, não obteve-se diferenças nos resultados.

PALAVRAS-CHAVE: Algoritmo Genético Multiobjetivo, Métricas C e S, Otimização de Portfólios.

Área principal : OA - Other applications in OR, GF - Financial Management, MH - Metaheuristics

ABSTRACT

This paper presents the application of multiobjective genetic algorithms (MOEAs) in the problem of portfolio optimization. The objective of this study was to evaluate the performance of genetic algorithms, NSGA-II and SPEA2, the investment problem into a set of actions. Was adopted to obtain the efficient frontier, the mean-variance model proposed by Markowitz (1952). The dataset was obtained from 26 active traded on BOVESPA period between 1 January 2004 to 31 December 2007. The algorithm performance was evaluated using the metric C and S. The result obtained by the metric C indicates that approximately 61% of the solutions obtained by NSGA-II are dominated by the solutions of SPEA. With respect to the metric S, no differences in the results obtained.

KEYWORDS: Multiobjective Genetic Algorithm, Metrics C and S, Portfolio.

Main Area : OA - Other applications in OR, GF - Financial Management, MH - Metaheuristics

1 Introdução

Carteira de investimento (portfólio) é um conjunto de ativos constituídos por um investidor, basicamente com dois objetivos: maximizar seus lucros e minimizar os riscos.

Para aplicar um capital numa carteira de ações deve-se primeiro a selecionar a carteira, isto é, deve-se escolher um conjunto de ações, dentre todas as que são negociadas na Bolsa de Valores, de modo que se tenha a maior perspectiva de lucro. Após a seleção, o problema seguinte consiste em determinar a quantidade ou a porcentagem do capital a ser investido em cada ativo selecionado. Para este fim deve-se considerar o retorno esperado de cada ativo e o risco que cada investidor está disposto a aceitar em seu investimento.

O problema de seleção de portfólios abordado neste trabalho consiste em determinar de que maneira deve ser aplicado um capital em um determinado conjunto de possíveis opções de investimento, sendo tais opções chamadas de ativos. O maior objetivo de um investidor é obter o máximo retorno (lucro), porém a este lucro está associado um risco. No processo de seleção de portfólios, o grande desafio é então encontrar relações que estabeleçam um bom compromisso entre retorno e risco.

Este problema pode ser resolvido por diferentes técnicas, como Programação quadrática, Simulated annealing, e Algoritmos evolutivos (Ehrgott, 2000).

A decisão ótima é dada por um vetor $x = (x_1, x_2, \dots, x_n)$, onde $x_i, i = 1, 2, \dots, n$, é o percentual do capital C a ser investido no ativo i , sendo n o número de ativos.

A teoria moderna de portfólios assume que, para um dado nível de risco, um investidor consciente deseja o máximo de retorno, e para um dado nível de retorno esperado, o investidor deseja minimizar o risco. Existem também investidores extremos que querem somente maximizar o retorno (desconsiderando o risco) ou minimizar o risco (desconsiderando o retorno). Em geral, podemos descrever as formulações dos problemas de otimização de portfólios através de cinco diferentes formulações, apresenta-se a seguir o modelo proposto por Markowitz (1952)

Maximizar o retorno e minimizar o risco

$$\begin{aligned} \text{Max } f(x) &= \sum_{i=1}^n l_i x_i \\ \text{Min } f(x) &= \sum_{i=1}^n \sum_{j=1}^n x_i x_j \sigma_{ij} \\ &\sum_{i=1}^n x_i = 1 \\ &x_i \geq 0 \end{aligned}$$

onde

σ_{ij} é a covariância entre os ativos i e j

l_i é o retorno do ativo i depois de um período de tempo

A avaliação dos riscos de uma carteira de ativos financeiros é frequentemente medido em termos das variações dos preços de tais ativos. Supondo que não haja dividendos pagos no período, a variação relativa de preços, ou retorno líquido simples de um ativo, é definido por

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}}$$

onde P_t é o preço de um ativo no instante t , normalmente um dia de negócio. Usualmente se expressa R_t em porcentagem, relativamente ao período (um dia, um mês, uma ano, etc) e é chamado de taxa de retorno.

Considerando que se tenha selecionado um conjunto de ativos para constituir o portfólio, o objetivo deste trabalho é avaliar o desempenho de dois algoritmos evolutivos, para ajuda na decisão de investimento em ativos de um portfólio. A avaliação final é feita pelo investidor, que se desejar o maior lucro independentemente do risco que está associado a esta escolha, ele selecionará opções que lhe propiciem isto, deixando de lado aquelas que, embora tenham menor risco, têm menor possibilidade de retorno.

2 Algoritmo Genético Multiobjetivo

Algoritmos Genéticos, assim como outros algoritmos evolutivos, têm seu mecanismo inspirado no processo de evolução natural. Algumas ideias fundamentais da evolução, tais como recombinação e seleção são utilizadas na construção de algoritmos robustos e que requerem um mínimo de informações sobre o problema (Deb, 2001). Em problemas com mais de uma função objetivo, onde a solução é dada pelo conjunto ótimo de Pareto (fronteira eficiente), os algoritmos genéticos multiobjetivo têm se destacado pela habilidade de encontrar uma boa aproximação da solução exata. Por esse motivo, um grande número de algoritmos têm sido propostos, como por exemplo o MultiObjective Genetic Algorithm (MOGA), o Niche-Pareto Genetic Algorithm (NPGA) e o Non-dominated Sorting Genetic Algorithm (NSGA) (Coelho, 2000). Uma nova geração de algoritmos genéticos multiobjetivo, introduziram o elitismo (Zitzler, 1999), dentre os mais citados estão o Strength Pareto Evolutionary Algorithm (SPEA, SPEA2)(Zitzler, 2001) e NSGAI (Deb et. al., 2002)

O princípio básico dos algoritmos genéticos assim como outros algoritmos evolutivos é o mesmo (Eiben, 2003). Resumidamente, dado um conjunto de funções a serem otimizadas:

1. Gera-se aleatoriamente um grupo de soluções candidatas. Estas soluções devem ser codificadas (cada solução passa a chamar então, indivíduo) definindo o genótipo, representação da solução no espaço de decisão, e o fenótipo, representação da solução no espaço dos objetivos. A codificação pode ser feita por representação binária, representação em ponto flutuante ou representação real.
2. Aplica-se uma função para medir a qualidade destas soluções candidatas (população), atribuindo-as um valor que mede sua adequação, chamado fitness.
3. Algumas das melhores soluções são selecionadas (pais), com base no fitness, a fim de originar uma nova população.
4. Operadores de recombinação e/ ou mutação são aplicados. A recombinação é um operador aplicado a duas ou mais soluções candidatas (pais) resultando em duas ou mais

soluções (filhos). A mutação transforma uma solução em outra substituindo genes nos cromossomos.

5. As novas soluções candidatas (descendentes) são avaliadas com base no fitness e competindo com as soluções da geração anterior, geram a nova população.

2.1 NSGA-II: Non-dominated Sorting Genetic Algorithm

O algoritmo NSGA-II foi proposto por Deb et. al. (2002) e utiliza o conceito de níveis, e ranks na definição das fronteiras de Pareto.

Uma população inicial P_0 de tamanho N é criada aleatoriamente. Esta população é classificada em diferentes níveis (fronteiras) de dominância. A primeira fronteira definida é não-dominada e a segunda fronteira somente é dominada pelos indivíduos da primeira fronteira, e assim sucessivamente. A cada indivíduo de cada fronteira é designado um valor de rank (fitness) com base na fronteira a que ele pertence. Aos indivíduos da primeira fronteira é dado um valor de fitness igual a 1, e aos indivíduos da segunda fronteira é atribuído um valor de fitness igual a 2 e assim por diante. Deseja-se a minimização do fitness (menores fitness estão mais próximos da fronteira).

Além do valor de fitness, um novo parâmetro chamado distância crowding é calculado para cada indivíduo. A distância de crowding é uma medida de quão próximo um indivíduo está de seus vizinhos. Uma distância média de crowding grande, irá resultar em melhor diversidade da população.

Os pais são selecionados da população usando a seleção do torneio binário com base na classificação (rank) e na distância de crowding (aglomeração). Um indivíduo é selecionado se o rank é menor do que o outro ou se a distância de crowding é maior do que a um outro (a distância de crowding é comparada somente se o rank de ambos os indivíduos são os mesmos).

A população selecionada gera filhos através de operadores de cruzamento e mutação. Uma população intermediária composta pela população atual e a população de filhos gerada é classificada novamente com base na relação de não-dominância. Os melhores indivíduos são selecionados segundo o operador de crowding e estes vão formar a população da próxima geração.

O algoritmo consiste nos seguintes passos:

Passo 1 Crie uma população inicial de pais P_t de tamanho N .

Passo 2 A população P_t é classificada de acordo com a relação de não-dominância. Identifique as diferentes fronteiras $F_i, i = 1, 2, \dots$

Passo 3 Calcule a distância de crowding. Execute o procedimento Crowding-sort ($F_i, <_c$) descrito a seguir. (para cada indivíduo da população é designado um valor segundo a distância de crowding)

Procedimento Crowding-sort ($F_i, <_c$)

Para cada fronteira F_i , seja n o número de indivíduos desta fronteira

- Inicialize com $F_i(d_j) = 0$, onde j é o j -ésimo indivíduo na fronteira F_i
- Para cada função objetivo m

- * Classifique os indivíduos na fronteira F_i baseado na função objetivo m , isto é:
 $I^m = \text{sort}(F_i, m)$
- * Designe distância infinita para os limites inferiores e superiores para cada indivíduo em F_i , isto é, $I(d_1) = \infty$ e $I(d_n) = \infty$.
- * Para $k = 2$ até $n - 1$

$$d_{I_k^m} = d_{I_k^m} + \frac{f_{I_k^m}^{m+1} - f_{I_k^m}^{m-1}}{f_m^{\max} - f_m^{\min}}$$

onde $f_{I_k^m}^m$ é o valor da m -ésima função objetivo do k -ésimo indivíduo em I .

Obs. a idéia básica por trás da distância de crowding é encontrar a distância euclidiana entre cada indivíduo da fronteira com base em seus m objetivos no hiperespaço m dimensional. Os indivíduos limites são sempre selecionados por possuem distância infinita.

Passo 4 Seleção. Escolha de pais para cruzamento e mutação. A escolha é feita usando torneio binário e o operador *crowded-comparison* \prec_n como descrito a seguir.

1. rank de não-dominância p_{rank} . Indivíduos na fronteira F_i terão seu rank como $p_{rank} = i$
2. distância de crowding $F_j(d_j)$
 - $p \prec q$ se
 - $p_{rank} < q_{rank}$
 - ou, se p e q pertencem à mesma fronteira F_i (ou seja, tem mesmo rank), então $F_i(d_p) > F_i(d_q)$ ou seja, a distância de crowding deve ser a maior.

Passo 6 Crie a população filhos Q_t de P_t usando operadores de crossover e mutação.

Passo 7 Chame $R_t = Q_t \cup P_t$ uma população intermediária. Classifique R_t segundo a relação de não dominância. A população P_{t+1} para a próxima geração é criada usando as N melhores soluções de R_t obtidas via operador de crowding.

2.2 SPEA2-Strenght Pareto Evolutionary Algorithm

No algoritmo SPEA2 (Zitzler, 2001), uma população inicial é gerada, um arquivo vazio é criado e o fitness da população calculado. A seleção dos indivíduos é feita pelo operador torneio binário. O novo arquivo é preenchido de acordo com o valor do fitness associado com cada indivíduo. Todos os indivíduos com fitness menor do que 1 completam o arquivo, e se o tamanho do arquivo for menor do que o pré-estabelecido, o arquivo é preenchido com indivíduos dominados da solução corrente. Se excede, alguns indivíduos são removidos pelo operador de truncamento.

O SPEA2 é uma versão atualizada de seu antecessor SPEA, pois foram eliminados potenciais pontos fracos incorporando ao algoritmo: um programa melhorado de atribuição de fitness, que considera a cada indivíduo na contagem, quantos indivíduos dominam e por quantos é dominado; uma técnica de estimação de densidade utilizando o vizinho mais próximo, que permite uma melhor orientação durante o processo de busca; um novo método de truncamento de arquivo que garante a preservação de soluções de fronteira. Finalmente, outra diferença do SPEA2 é que somente membros do arquivo participam do processo de seleção de cruzamento.

O Algoritmo consiste nos seguintes passos:

- Passo 1** Inicialização: gerar uma população inicial P_0 e criar uma arquivo vazio A_0 (conjunto externo). Conjunto $t = 0$
- Passo 2** Atribuição de fitness: calcular os valores de fitness dos indivíduos em P_t e A_t
- Passo 3** Seleção natural: copiar todos os indivíduos não dominados em P_t e A_t para A_{t+1} . Se o tamanho exceder N então reduzir A_{t+1} utilizando o operador de truncamento, caso o contrário se o tamanho de A_{t+1} for menor do que N , então preencha A_{t+1} com indivíduos dominados em P_t e A_t .
- Passo 4** Finalização: se $t = T$ ou outro critério de parada for satisfeito, defina A^* para o conjunto de vetores de decisão representados pelos indivíduos não dominados em A_{t+1} . Pare.
- Passo 5** Seleção de Cruzamento: Realize seleção por meio de torneio binário com substituição em A_{t+1} a fim de preencher o conjunto de cruzamento.
- Passo 6** Variação: Aplique operadores de mutação e recombinação no conjunto de cruzamento e defina P_{t+1} para a população resultante. Incremente o contador de geração ($t = t + 1$) e vá para o passo 2.

2.3 Métricas

No intuito de comparar os diferentes métodos de otimização multiobjetivo diversas métricas foram criadas. A métrica C , proposta Deb (2001), avalia a proporção de soluções obtidas em um método que são dominadas por soluções obtidas em outro. Desta forma dados dois conjuntos A e B de soluções obtidas por algoritmos diferentes, a métrica C calcula a proporção de soluções em B , que são fracamente dominadas por soluções de A e é dada por,

$$C(A, B) = \frac{|\{b \in B | \exists a \in A : a \succ = b\}|}{|B|}$$

Para avaliar a diversidade de um conjunto não dominado de soluções, Schott (1995) sugere a métrica *Spacing* (S), que considera a distância relativa entre soluções consecutivas em um conjunto de pontos não dominados, dada por:

$$S = \sqrt{\frac{1}{|Q|} \sum_{i=1}^{|Q|} (d_i - \bar{d})^2}$$

onde $d_i = \min_{k \in Q \wedge k \neq i} \sum_{m=1}^M |f_m^i - f_m^k|$ é o valor mínimo dado pela soma da diferença absoluta dos valores da função objetivo, entre a i -ésima solução e outra qualquer pertence ao conjunto. O valor \bar{d} é a média das distâncias definidas acima, Q é o número de soluções no conjunto não dominado e M é o número de funções objetivo.

A métrica S , mede os desvios padrões dos diferentes d_{is} . Quando as soluções são próximas e uniformemente espaçadas o correspondente valor medido é menor. Logo, um algoritmo que encontra um menor valor de S , para o conjunto não dominado é melhor.

3 Aplicação Numérica e Resultados

O modelo apresentado nas equações 1 e 2 do item 1, foi resolvido pelos algoritmos genéticos NSGA-II e SPEA2, e obtidas as respectivas fronteiras eficientes. A fronteira exata foi obtida pelo método dos pesos (Goicoechea, 1982). Os dados empregados nos experimentos, foram extraídos de um conjunto de 26 ativos mais negociados no BOVESPA entre período 01 de janeiro de 2004 a 31 de dezembro de 2007.

Na tabela 1 estão os parâmetros utilizados nos experimentos. O comprimento do gene é igual ao número de ativos, e o critério de parada dos algoritmos foi o número máximo de 200 iterações.

Tabela 1: Parâmetros para NSGA-II e SPEA2

	NSGA-II	SPEA2
tamanho da população	100	100
número de iterações	100	50
probabilidade de crossover	0.9	0.9
probabilidade de mutação	0.1	0.1

Os resultados foram obtidos usando o Matlab 7.9.0 (R2009b), em um computador com processador Intel(R)Atom(TM) 1,60GHz de CPU e 1Gb de RAM.

As fronteiras eficientes geradas pelos algoritmos SPEA2 e NSGA-II, exibidas nas figuras 1 e 2 respectivamente, são comparadas com a fronteira exata. O desempenho dos AGs foi comparado por meio da avaliação das métricas C e S estão apresentados nas tabelas 2 e 3. Os resultados obtidos pela métrica C, apontam que 61,54 % das soluções obtidas pela NSGA, são dominadas pelas soluções obtidas pelo SPEA, e 7% das soluções do SPEA são dominadas pelas do NSGA. Para a métrica S, não houve diferença entre os valores obtidos.

Tabela 2: Resultados obtidos pela métrica C

	NSGA-II	SPEA2
NSGA-II	–	0,61538
SPEA2	0,07000	–

Tabela 3: Resultados obtidos pela métrica S

NSGA-II	$1,88 \times 10^{-6}$
SPEA2	$1,77 \times 10^{-6}$

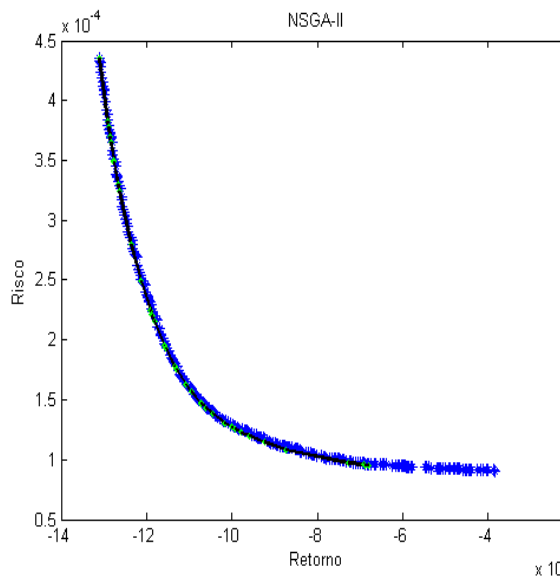


Figura 1: Fronteira Eficiente - NSGA-II

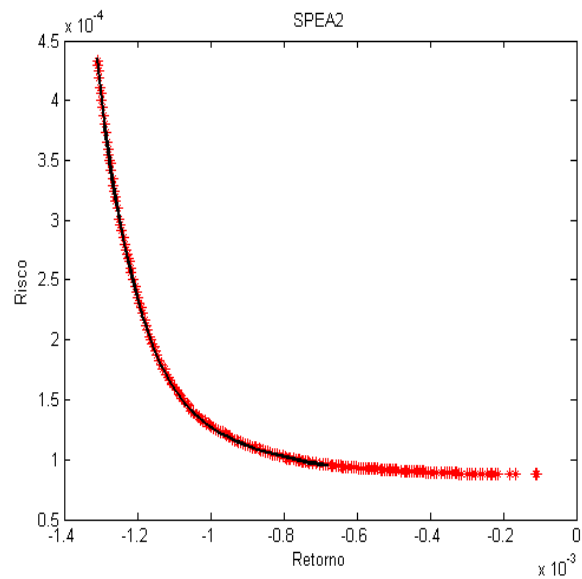


Figura 2: Fronteira Eficiente - SPEA2

4 Conclusões

O estudo de algoritmos evolutivos aplicados em um problema de otimização multiobjetivo de portfólios é o principal foco deste trabalho. Foram utilizados dois algoritmos genéticos multiobjetivos para obtenção da fronteira Pareto-Ótima, o NSGA-II e SPEA2. Com relação à convergência, avaliada por meio da métrica C, o SPEA2 apresenta um grande número de soluções que são melhores do que as obtidas pelo NSGA. Em ambos os algoritmos a fronteira obtida é bastante diversificada, e os resultados obtidos pela métrica S, não diferem significativamente. Futuros trabalhos introduzirão diferentes operadores de busca local nos modelos existentes com os quais será possível uma melhor exploração do espaço de busca quando aplicado em problemas desta natureza.

Referências

- Deb, K.** Multi-objective optimization using evolutionary algorithms. *John Wiley and Sons*, 2001.
- Coello, C.A.C.** An updated survey of GA-based multiobjective optimization techniques. *ACM Computing Surveys*, 32(2), 109-143, 2000.
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.** A fast and elitist multiobjective genetic algorithm: NSGA II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182-197, 2002.
- Ehrgott, M., Gandibleux, X.** A Survey and Annotated Bibliography on Multiobjective Combinatorial Optimization. *OR Spectrum*, 2000.

- Eiben, A.E., Smith, J.E.** Introduction to Evolutionary Computing *Springer*, 2003.
- Goicoechea, A., Hansen, D.** Multiobjective decision analysis with engineering and business applications. *John Wiley Sons*, 1982.
- Markowitz, H.M.** Portfolio selection. *Journal of Finance*, 77-91, 1952.
- Schott, J.R.** Fault tolerant design using single and multi-criteria genetic algorithms. Master's Thesis, Massachusetts Institute of Technology, 1995.
- Zitzler, E., Thiele, L.** Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4), 257-271, 1999.
- Zitzler, E. and Deb, K. and Thiele, L. and Coello, C. A. and Corne, D.** Evolutionary Multi-Criterion Optimization (EMO 2001). *Springer*, 2001.