



Uma Nova Abordagem Baseada em Algoritmo Evolutivo Multiobjeto Aplicado no Problema do Caixeiro Viajante Biobjetivo

Deyvid Heric de Moraes

Universidade Tecnológica Federal do Paraná-UTFPR
Av. Alberto Carazzai, 1640 - Centro, Cornélio Procópio - PR, 86300-000
moraes.deyvid@gmail.com

Danilo Sipoli Sanches

Universidade Tecnológica Federal do Paraná-UTFPR
Av. Alberto Carazzai, 1640 - Centro, Cornélio Procópio - PR, 86300-000
danilosanches@utfpr.edu.br

Josimar da Silva Rocha

Universidade Tecnológica Federal do Paraná- UTFPR
Av. Alberto Carazzai, 1640 - Centro, Cornélio Procópio - PR, 86300-000
jrocha@utfpr.edu.br

RESUMO

Neste trabalho é apresentado uma nova abordagem de algoritmo evolutivo multiobjetivo, denominado MOEA/NSM (do Inglês, *Multiobjective Evolutionary Algorithm integrating NSGA-II, SPEA2 and MOEA/D features*). O algoritmo preserva em geral, características de um algoritmo evolutivo, concentrando qualidades de outras abordagens de sucesso na literatura em uma única abordagem, para que elas trabalhem em conjunto, por meio de subpopulações. O objetivo do estudo foi combinar as principais características dos algoritmos NSGA-II, SPEA2 e MOEA / D, e incluir ainda uma técnica de busca local para melhorar a busca no espaço de objetivos. o algoritmo MOEA/NSM foi comparado as demais abordagens clássicas utilizando 9 *datasets* para o problema do caixeiro viajante biobjetivo. A partir dos resultados obtidos foram aplicados a técnica de avaliação por Hipervolume e um teste de hipótese estatístico. Mais detalhes serão apresentados neste trabalho.

PALAVRAS CHAVE. Algoritmos Evolutivos Multiobjetivo, Problema do Caixeiro Viajante Biobjetivo, NSGA-II, SPEA2, MOEA/D, Busca Local.

Apoio à Decisão Multicritério, Metaheurísticas, Otimização Combinatória.

ABSTRACT

This paper presents a multiobjective evolutionary algorithm approach, called MOEA/NSM. The proposed approach is based on subpopulations with local search, to deal with the biobjective traveling salesman problem. The algorithm preserves in general, characteristics of an evolutionary algorithm, concentrating qualities of other approaches of success in the literature in a single, so that they work together. The purpose of the study was to combine the qualities of the NSGAII, SPEA2 and MOEA / D algorithms, to apply local search and to compare the results obtained through the metric Hypervolume and statical test.

KEYWORDS. Multi-Objective Evolutionary Algorithms, Bi-Objective Traveling Salesperson Problem, NSGA-II, SPEA2, MOEA/D, Local Search.

Multi Objective optimization, Metaheuristics, Combinatorial Optimization.



1. Introdução

Problemas do Caixeiro Viajante com m -objetivos e n cidades são geralmente representados por m grafos ponderados, onde, para cada $k \in \{1, \dots, m\}$, o grafo G_k é um grafo ponderado que representa o k -ésimo objetivo cujos vértices representam as cidades e são indicados por números inteiros do conjunto $Y = \{0, 1, \dots, n-1\}$ e $d_{ij}^{(k)}$ representa o custo (ou peso) da aresta com extremidades nos vértices i e j de G_k , ou seja, o custo em G_k de percorrer o caminho da cidade representada pelo vértice i para a cidade representada pelo vértice j sem percorrer cidades intermediárias. Observe que para objetivos diferentes o custo poderá representar distância, tempo, gastos ou outra informação de interesse. Neste trabalho, trabalharemos com o Problema do Caixeiro Viajante Simétrico em que $d_{ij}^{(k)} = d_{ji}^{(k)}$ para todos $i, j \in Y$.

Matematicamente, o Problema do Caixeiro Viajante com m objetivos e n cidades utilizado neste trabalho consiste em determinar os valores de variáveis binárias x_{ij} definidas por

$$x_{ij} = \begin{cases} 1, & \text{se } (i, j) \text{ é uma aresta utilizada} \\ & \text{no caminho percorrido} \\ 0, & \text{caso contrário} \end{cases}$$

e satisfazendo

$$\sum_{j=0}^{n-1} x_{ij} = 1$$

e

$$\sum_{i=0}^{n-1} x_{ij} = 1.$$

de tal forma a minimizar o vetor (y_1, \dots, y_m) , onde, para cada $k = 1, \dots, m$,

$$y_k = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} x_{ij} d_{ij}^{(k)},$$

ou equivalentemente, se $Y = \{0, 1, \dots, n-1\}$ e Ω é o espaço de busca formado por todos os elementos de Y^n que possuem todas as coordenadas distintas, então o problema do Caixeiro Viajante com m objetivos e n cidades consiste em encontrar $\tilde{x} \in \Omega$ de tal forma que

$$f(\tilde{x}) = \min_{x \in \Omega} f(x),$$

onde

$$f(x) = (f_1(x), f_2(x), \dots, f_m(x))$$

e, para cada $k \in \{1, 2, \dots, m\}$ e $x = (x_0, x_1, \dots, x_{n-1}) \in \Omega$,

$$f_k(x) = f_k(x_0, \dots, x_{n-1}) = \sum_{i=0}^{n-2} d_{x_i x_{i+1}}^{(k)} + d_{x_{n-1} x_0}^{(k)}.$$

O Problema do Caixeiro Viajante multiobjetivo (TSPMO) tem larga aplicabilidade em problemas de logística que envolvam transporte de mercadorias ou de serviços em que tentam alcançar diversos objetivos simultaneamente, tais como minimização de distâncias percorridas, do custo do transporte, do tempo e maximização da quantidade de mercadorias distribuídas, entre outros.

Como o Problema do Caixeiro Viajante (TSP) com mais de um objetivo é considerado um problema NP-Difícil [Psychas et al., 2015], muitos pesquisadores buscaram utilizar este problema para testar novas abordagens, como por exemplo, o algoritmo baseado em duas fases [Paquete e



Stützle, 2003], o método 2PPLS [Lust e Teghem, 2010b] onde foi aplicado técnicas de aceleração ao método antes proposto por [Lust e Teghem, 2007], uma análise da aplicação de métodos de busca local ao TSP multiobjetivo [Paquete e Stützle, 2009], busca tabu [Hansen, 2000], algoritmos genéticos [Elaoud et al., 2010], algoritmos evolutivos baseado em decomposição (MOEA/D) [Peng et al., 2009], algoritmos evolutivos híbridos [Kumar e Singh, 2007; Psychas et al., 2015; Köksalan e Öztürk, 2016], otimização por colônia de formigas [Angus, 2007; García-Martínez et al., 2007; Zhang e Li, 2007], algoritmos meméticos [Krasnogor e Smith, 2000; Jaszkievicz, 2002; Jaszkievicz e Zielniewicz, 2009; Samanlioglu et al., 2008; Cornu et al., 2016], entre outros. Para uma visão mais detalhada do TSP com mais de um objetivo consultar [Lust e Teghem, 2010a].

Dada à dificuldade de resolver problemas de otimização multiobjetivo (MOOPs), a aplicação de técnicas de aproximação têm sido largamente estudadas nas últimas décadas. Os algoritmos evolutivos (EAs) são promissores para serem empregados em MOOPs em razão de apresentarem as seguintes características: trabalhem com mais de uma função simultaneamente; não precisarem de informações adicionais e serem capazes de escapar de ótimos locais [Sanchez, 2013]. Nesse sentido, existe uma necessidade de se estudar novos algoritmos evolutivos, buscando sempre por melhores resultados.

Algumas abordagens baseadas na decomposição do problema em subproblemas apresentaram ótimos resultados tanto para problemas de otimização contínua utilizando funções benchmark [Zhang e Li, 2007] como em problemas reais de otimização discreta (MEA2N) [Gois et al., 2013]. No MEA2N foi implementado a representação por nó-profundidade e destaca-se por utilizar de subpopulações, onde armazena as melhores soluções encontradas de acordo com os diferentes critérios para cada objetivo do problema. Esta abordagem inclui características de ordenação por não dominância dos algoritmos NSGA-II e SPEA2.

Com base no bom desempenho do MEA2N, neste trabalho foi utilizado um conceito semelhante, como principal característica a decomposição do problema em subpopulações, sendo que para cada subpopulação foi utilizado uma medida de distância diferente baseada nos algoritmos SPEA2, NSGA-2 e MOEA/D. Utilizando as técnicas de truncamento utilizado no SPEA2 e a distância de multidão utilizada no NSGA-II, é possível melhorar a convergência e diversidade na fronteira de Pareto em relação ao uso separado destas duas abordagens. Além disso, a abordagem proposta faz uso da função de agregação de Tchebycheff utilizada no MOEA/D e uma técnica de busca local para melhorar a qualidade das soluções no espaço de busca. Diante disto, o objetivo deste trabalho é comparar a nova abordagem proposta, chamada aqui de MOEA/NSM, com algoritmos do estado da arte aplicados ao BOTSP e analisar os resultados a partir da métrica hipervolume e teste de hipótese estatístico.

2. Método proposto

Nas últimas décadas muitos algoritmos foram propostos para resolução de MOOPs, dentre os mais conhecidos estão os algoritmos NSGA-II [Deb et al., 2000], SPEA2 [Zitzler et al., 2001] e o MOEA/D [Zhang e Li, 2007]. Diante disto, neste trabalho é apresentado uma abordagem que se baseia em subpopulações, incluindo características utilizadas nesses algoritmos e empregando o uso da busca local. A Figura 1 apresenta a estrutura da abordagem utilizada. A abordagem utiliza a distância de multidão proposta no NSGA-II, o truncamento proposto no SPEA2 e o método de Tchebycheff usado no MOEA/D como meio de busca da diversidade da população.

Na abordagem proposta é tratado o Problema do Caixeiro Viajante Biobjetivo (BOTSP) cujas funções objetivos são obtidas por cada uma das 9 instâncias do banco de dados disponível em <https://eden.dei.uc.pt/~paquete/tsp/#Exp2>, sendo estas: euclidAB100, euclidCD100, euclidEF100, euclidAB300, euclidCD300, euclidEF300, euclidAB500, euclidCD500 e euclidEF500.

Para cada uma das instâncias abordadas são utilizadas subpopulações, onde cada subpopulação tem uma função diferente da outra. As populações $P_t^{obj^1}$ e $P_t^{obj^2}$ armazenam soluções para cada objetivo que são preenchidas inicialmente de forma aleatória, mas que no decorrer das

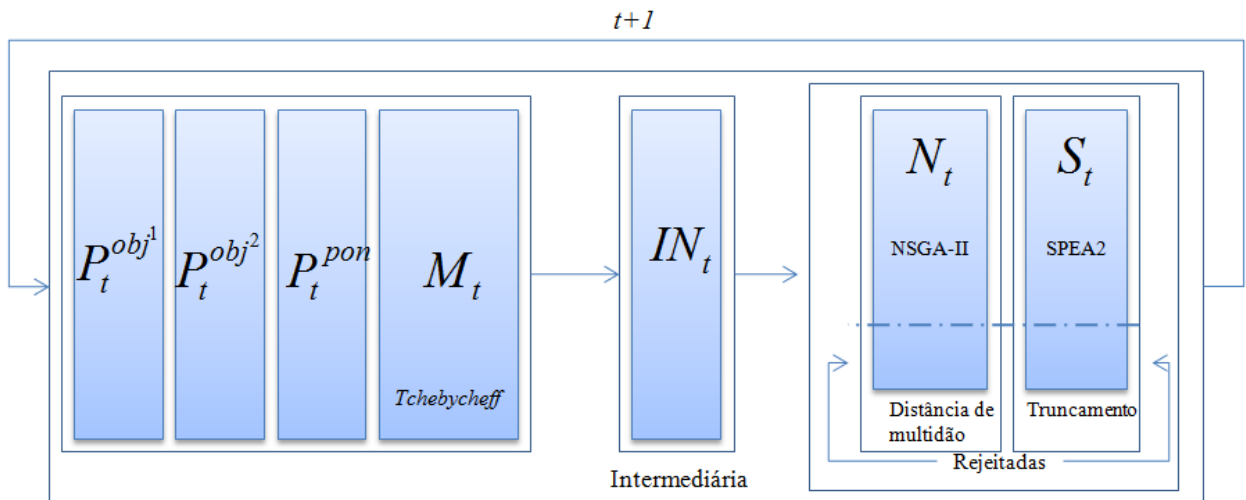


Figura 1: Método Proposto

iterrações buscam sempre a minimizaçaõ das soluções. A populaçaõ P_t^{pon} armazena uma soma de todos os n objetivos para cada soluçaõ com a finalidade de obter pontos na regiõ da fronteira conhecida por joelho. A busca local foi essencial para encontrar boas soluções para essas subpopulações. A Figura 2 justifica a necessidade dessas subpopulações, levando em consideraçaõ que sãõ áreas que oferecem grande dificuldade para serem exploradas.

A subpopulaçaõ M_t tem a responsabilidade de proporcionar a diversidade, para isto foi implementado a funçaõ de agregaçaõ de *Tchebycheff*. O mètodo é apresentado a seguir:

$$\text{Minimizar } g^{te}(x|\lambda, z^*) = \max_{1 \leq i \leq m} \{\lambda_i |f_i(x) - z_i^*|\} \quad (1)$$

sujeito a $x \in \Omega$

Tal que $z^* = (z_1^*, \dots, z_m^*)^T$ é o ponto de referênciã, baseando-se no melhor valor de aptidãõ encontrado até i , $z_i^* = \min \{f_i(x) | x \in \Omega\}$ para cada $i = 1, \dots, m$. Para cada ponto ótimo x^* existe um vetor de pesos λ na qual x^* é a soluçaõ ótima da equaçãõ 1 que por sua vez é uma soluçaõ ótima de Pareto.

Seja $\lambda^1, \dots, \lambda^{Sp}$ um conjunto de vetores de pesos e z^* o ponto de referênciã, a aproximaçaõ da fronteira de Pareto de um MOOP pode ser decomposto em S^p problemas mono-objetivo utilizando a funçaõ de agregaçaõ. O conjunto de pesos λ tem a funçaõ de manter a diversidade das soluções. A relaçaõ de vizinhança entre estes subproblemas sãõ definidos com base nas distânciãs entre seus vetores de pesos de agregaçaõ. Cada subproblema é otimizado usando informações principalmente de seus subproblemas vizinhos.

Em M_t ainda sãõ aplicados o cruzamento e a busca local aleatõria. Na busca local aleatõria é realizado um sorteio e por meio probabilístico, um dos objetivos ou a soma ponderada pode ser selecionado para minimizaçaõ.

Inicialmente criamos as populações P_t^{obj1} , P_t^{obj2} , P_t^{pon} , M_t , S_t e N_t aleatoriamente.

Aplicando o torneio ternário sob as populações P_t^{obj1} , P_t^{obj2} , P_t^{pon} , M_t , S_t e N_t , comparamos os indivíduos por grau de dominânciã em que os indivíduos dominados por menos indivíduos vencem o torneio e sãõ submetidos ao cruzamento ou busca local. Todas as novas soluções geradas aplicando-se o cruzamento e a busca local formarãõ a subpopulaçaõ intermediária IN_t .

Ao fim de cada ciclo, criamos as subpopulações P_{t+1}^{obj1} , P_{t+1}^{obj2} e P_{t+1}^{pon} a partir das melho-

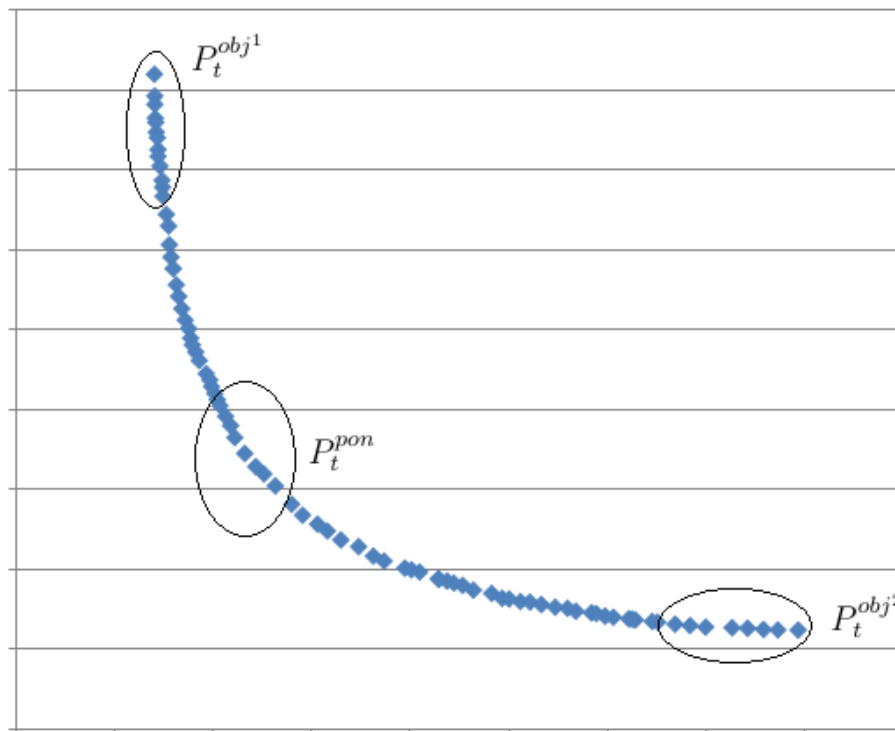


Figura 2: Pontos de difícil acesso encontrados pela busca local

res soluções obtidas entre $P_t^{obj^1}$ e IN_t , $P_t^{obj^2}$ e IN_t e P_t^{pon} e IN_t , respectivamente, conforme a função objetivo e respeitando o tamanho da população correspondente. Para a obtenção das melhores soluções ordenamos as soluções por ordem de aptidão conforme a função objetivo até alcançar o tamanho da população desejada de forma que os indivíduos excedentes são descartados. Para a obtenção da subpopulação M_{t+1} utilizamos a função de agregação de Tchebyscheff a concatenação das populações M_t e IN_t e descartamos os indivíduos menos aptos. A criação da população N_{t+1} é obtida a partir da concatenação das listas de indivíduos de N_t e IN_t ordenados por fronteiras de dominância F_1, F_2, F_3, \dots , e utilizamos a distância de multidão para eliminar o excesso de indivíduos quando a quantidade máxima de indivíduos de N_t for alcançada. A distância de multidão usa uma estimativa de densidade das soluções vizinhas de cada solução da população. Baseando-se nessa vizinhança, calcula-se a média da distância das duas soluções adjacentes (cubóide) de cada indivíduo para todos os objetivos. Já para a criação da subpopulação S_{t+1} , concatenamos os indivíduos das populações S_t e de IN_t em cada indivíduo recebe um valor de aptidão e é comparado com todos os outros para aplicação do truncamento. O truncamento, remove soluções cuja distância para o seu vizinho mais próximo seja a menor dentre as soluções existentes, no caso de empate, calcula-se a segunda menor distância, e assim por diante. Inicialmente algumas soluções podem estar bem próximas, após a aplicação do algoritmo essas soluções são removidas preservando as soluções extremas de cada objetivo, garantindo a diversidade da população.

O operador de cruzamento utilizado foi o *Order Crossover* (OX) proposto por [Davis, 1985]. O mesmo operador foi utilizado para todas as abordagens estudadas.

Com base nos conceitos apresentados, algumas combinações de métodos foram construídos com o objetivo de observar o comportamento de cada método dentro da abordagem utilizada.

O algoritmo 1 descreve as etapas da abordagem proposta detalhadamente e o algoritmo 2 descreve a busca local utilizada.



Algoritmo 1: Funcionamento do algoritmo proposto

Entrada: $maxGer$
Saída: $fronteira$

1 início
2 $P_1^{obj1}, P_1^{obj2}, P_1^{pon}, M_1, S_1, N_1 \leftarrow criarPopInicialAleatoria();$
3 para $t \leftarrow 1, 2, \dots, maxGer - 1$ **faça**
4 $individuo[1] = aplicarSelecao(P_t^{obj1}, P_t^{obj2}, P_t^{pon}, M_t, S_t, N_t);$
5 $individuo[2] = aplicarSelecao(P_t^{obj1}, P_t^{obj2}, P_t^{pon}, M_t, S_t, N_t);$
6 $aplicarCruzamento(individuo);$
7 $aplicarBuscaLocal(individuo);$
8 $avaliarAptidao(individuo);$
9 $IN_t \leftarrow individuo;$
10 $P_{t+1}^{obj1} \leftarrow aplicarInsercaoPorAptidao(P_t^{obj1}, IN_t);$
11 $P_{t+1}^{obj2} \leftarrow aplicarInsercaoPorAptidao(P_t^{obj2}, IN_t);$
12 $P_{t+1}^{pon} \leftarrow aplicarInsercaoPorAptidao(P_t^{pon}, IN_t);$
13 $M_{t+1} \leftarrow aplicarInsercaoTchebycheff(M_t, IN_t);$
14 $S_{t+1} \leftarrow aplicarTruncamento(S_t + IN_t);$
15 $N_{t+1} \leftarrow apliDistMultidao(aplicarRanking(N_t + IN_t));$
16 fim
17 $fronteira = aplicarRanking(S_{t+1} + N_{t+1});$
18 return $fronteira;$
19 fim

3. Resultados

Os algoritmos foram desenvolvidos em linguagem de programação Java contando com o auxílio da lib JMetal [Durillo e Nebro, 2011]. O ambiente de simulação foi em um computador com processador Core i7, com 16 Gbytes de memória RAM, sistema operacional Windows 7. Foram executados 30 experimentos de forma independente. O critério de parada foi quantidade de avaliações com o valor máximo de 1.000.000.

O número de indivíduos de cada subpopulação da abordagem adotada foi de: 20 indivíduos para cada objetivo (P^{obj1} e P^{obj2}); 40 para a subpopulação responsável pela média ponderada (P^{pon}); 100 para subpopulação Tchebycheff (M) e de 50 para as subpopulações controladas pelo NSGA-II (N) e SPEA2 (S). Para comparação com outros algoritmos da literatura, foi atribuída uma população de 100 indivíduos para cada abordagem: MOEA/D, NSGA-II e SPEA2. O vetor de pesos utilizado para o MOEA/D foi de 100 uniformemente distribuídos.

A comparação de desempenho de um ou vários métodos de otimização multi-objetivo é uma tarefa complexa. Segundo Deb [2001] duas metas da otimização multi-objetivo são: convergência e diversidade das soluções encontradas. Para tanto, segundo Deb [2001] existem diversas métricas que ajudam na análise destas metas com destaque para as métricas:

- Taxa de Erro;
- Distância Geracional;
- Métrica de Cobertura;
- Espaçamento;
- Espalhamento;



Algoritmo 2: Busca Local 2-opt

Entrada: *solucao*, *qtdTrocasPermitidas*, *matrixDist*

Saída: *solucao*

```
1 início
2   contador ← 0 ;
3   para i ← 0, 1, ..., (solucao - 2) faça
4     para j ← i + 2, ..., (solucao - 2) faça
5       troca ←
6         matrixDist[solucao[i], solucao[j]] + matrixDist[solucao[i +
7           1], solucao[j + 1]] - matrixDist[solucao[i], solucao[i + 1]] +
8           matrixDist[solucao[j], solucao[j + 1]] ;
9       se troca < 0 então
10        realizarTroca(i, j, solucao);
11        se contador == qtdTrocasPermitidas então
12          return fim;
13        fim
14      senão
15        contador = contador + 1 ;
16      fim
17    fim
18  fim
```

Entretanto, atualmente uma métrica bastante usada na avaliação de algoritmos multi-objetivos é o indicador de hipervolume (HV) Deb [2001]. No HV, calcula-se o volume da região coberta entre os pontos das soluções na fronteira de Pareto P e um ponto de referência W . Para cada solução i pertencente a P , é construído um hipercubo com volume v_i com referência a um ponto W . O ponto de referência pode ser encontrado construindo-se um vetor com os piores valores da função objetivo. O resultado da métrica é a soma dos volumes de todos os hipercubos encontrados. Quanto maior o valor do HV melhor, pois um alto valor de HV indica que houve um elevado espalhamento entre as soluções de P e indica que houve também uma melhor convergência. O HV é calculado da seguinte maneira:

$$HV = \sum_{i \in P} v_i \quad (2)$$

Diante disto, os experimentos foram realizados com 9 instâncias do BOTSP utilizando a métrica HV. As instâncias foram obtidas no banco de dados disponível em <https://eden.dei.uc.pt/~paquete/tsp/#Exp2>, sendo estas: euclidAB100, euclidCD100, euclidEF100, euclidAB300, euclidCD300, euclidEF300, euclidAB500, euclidCD500 e euclidEF500.

Os resultados foram obtidos a partir da execução de 30 experimento com cada abordagem para cada dataset. Os resultados obtidos estão ilustrados nas figuras 3(a) a 5(c). Diante disto, é possível observar um melhor desempenho da abordagem MOEA/NSM em relação as demais abordagens, uma vez que, para todos os datasets avaliados, a abordagem proposta obteve um melhor valor de hipervolume (próxima a 1).

A partir dos resultados obtidos foi aplicado um teste estatístico para verificar o desempenho da abordagem MOEA/NSM em relação as outras abordagens utilizadas. Diante disto, o teste de Shapiro–Wilk foi aplicado a cada conjunto de parâmetros. No caso da normalidade ser verifi-

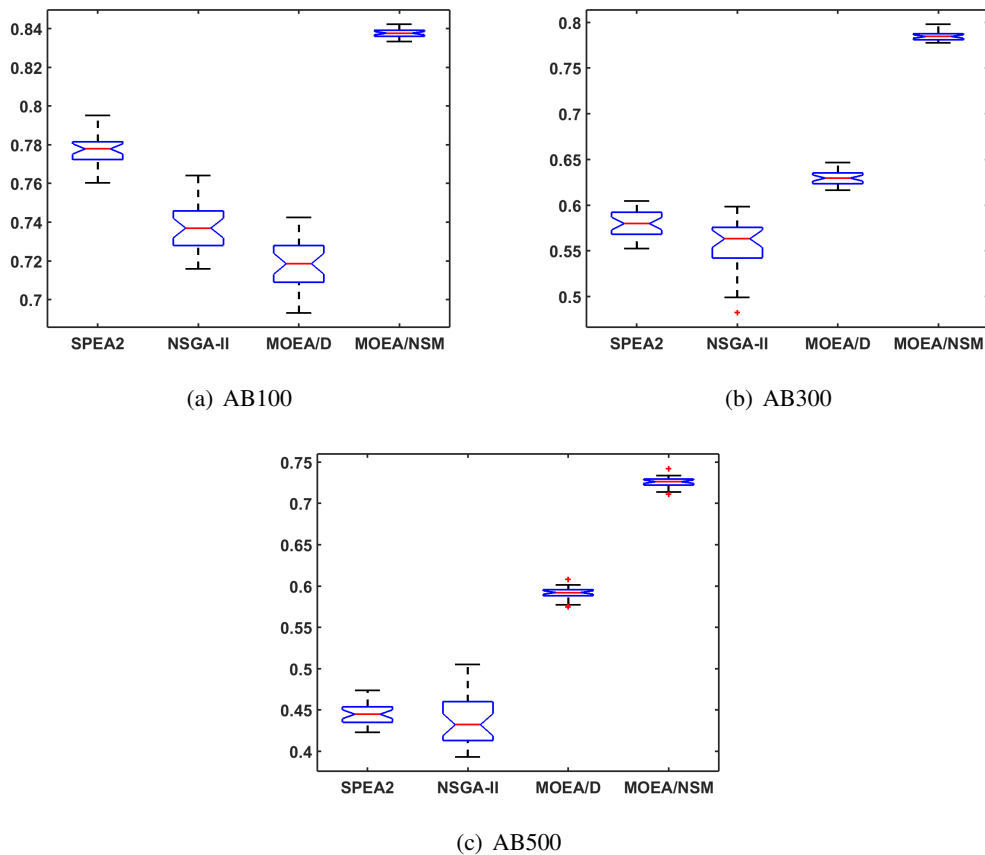


Figura 3: Valores de HV para os datasets AB100, AB300 e AB500.

cada, um teste t de Student pareado foi aplicado, caso contrário, o teste não paramétrico Wilcoxon pareado foi utilizado. Além disso, o nível de significância utilizado foi de 0.05 ou 95%.

Para os 9 datasets utilizados, a abordagem proposta apresentou diferença estatística significativa (p -valor $< 0,005$) (ver Tabela 1), indicando o bom desempenho a partir da integração das técnicas de distâncias utilizadas nas abordagens NSGA-2, SPEA2 e MOEA/D.

4. Conclusões

Neste trabalho é apresentado uma abordagem chamada MOEA/NSM que é baseada em subpopulações que utilizam características dos algoritmos NSGA-2, SPEA2 e MOEA/D. Além disso, foi utilizado também uma técnica de busca local 2-opt para melhorar a busca de soluções vizinhas no espaço de busca. A abordagem utiliza a distância de multidão proposta no NSGA-II, o truncamento proposto no SPEA2 e o método de Tchebycheff usado no MOEA/D como meio de busca da diversidade da população.

Além disso, na abordagem proposta se faz o uso de subpopulações, onde cada subpopulação tem uma função diferente da outra. As populações $P_t^{obj^1}$ e $P_t^{obj^2}$ armazenam soluções para cada objetivo que são preenchidas inicialmente de forma aleatória, mas que no decorrer das iterações buscam sempre a minimização das soluções. A população P_t^{pon} armazena uma soma de todos os n objetivos para cada solução com a finalidade de obter pontos na região da fronteira conhecida por joelho.

O processo de validação da abordagem proposta foi realizado utilizando 9 datasets e a métrica por hipervolume. Além disso, foi aplicado um teste estatístico para verificar o desempenho da abordagem proposta em relação as outras abordagens utilizadas. Para os 9 datasets utilizados, a abordagem proposta apresentou diferença estatística significativa (p -valor $< 0,005$), indicando o



Tabela 1: p-valor obtido para os 9 datasets avaliados

		MOEA/NSM	MOEA/D	NSGA-II	SPEA2
AB100	SPEA2	0,0001	0,0001	0,0001	#
	NSGA-II	0,0001	0,0001	#	0,0001
	MOEA/D	0,0001	#	0,0001	0,0001
		MOEA/NSM	MOEA/D	NSGA-II	SPEA2
AB300	SPEA2	0,0001	0,0001	0,0001	#
	NSGA-II	0,0001	0,0001	#	0,0001
	MOEA/D	0,0001	#	0,0001	0,0001
		MOEA/NSM	MOEA/D	NSGA-II	SPEA2
AB500	SPEA2	0,0001	0,0001	0,0655	#
	NSGA-II	0,0001	0,0001	#	0,0001
	MOEA/D	0,0001	#	0,0001	0,0001
		MOEA/NSM	MOEA/D	NSGA-II	SPEA2
CD100	SPEA2	0,0001	0,0001	0,0001	#
	NSGA-II	0,0001	0,0001	#	0,0001
	MOEA/D	0,0001	#	0,0001	0,0001
		MOEA/NSM	MOEA/D	NSGA-II	SPEA2
CD300	SPEA2	0,0001	0,0001	0,0001	#
	NSGA-II	0,0001	0,0001	#	0,0001
	MOEA/D	0,0001	#	0,0001	0,0001
		MOEA/NSM	MOEA/D	NSGA-II	SPEA2
CD500	SPEA2	0,0001	0,0001	0,0496	#
	NSGA-II	0,0001	0,0001	#	0,0001
	MOEA/D	0,0001	#	0,0001	0,0001
		MOEA/NSM	MOEA/D	NSGA-II	SPEA2
EF100	SPEA2	0,0001	0,0001	0,0001	#
	NSGA-II	0,0001	0,0001	#	0,0001
	MOEA/D	0,0001	#	0,0001	0,0001
		MOEA/NSM	MOEA/D	NSGA-II	SPEA2
EF300	SPEA2	0,0001	0,0001	0,0085	#
	NSGA-II	0,0001	0,0001	#	0,0001
	MOEA/D	0,0001	#	0,0001	0,0001
		MOEA/NSM	MOEA/D	NSGA-II	SPEA2
EF500	SPEA2	0,0001	0,0001	0,9989	#
	NSGA-II	0,0001	0,0001	#	0,0001
	MOEA/D	0,0001	#	0,0001	0,0001

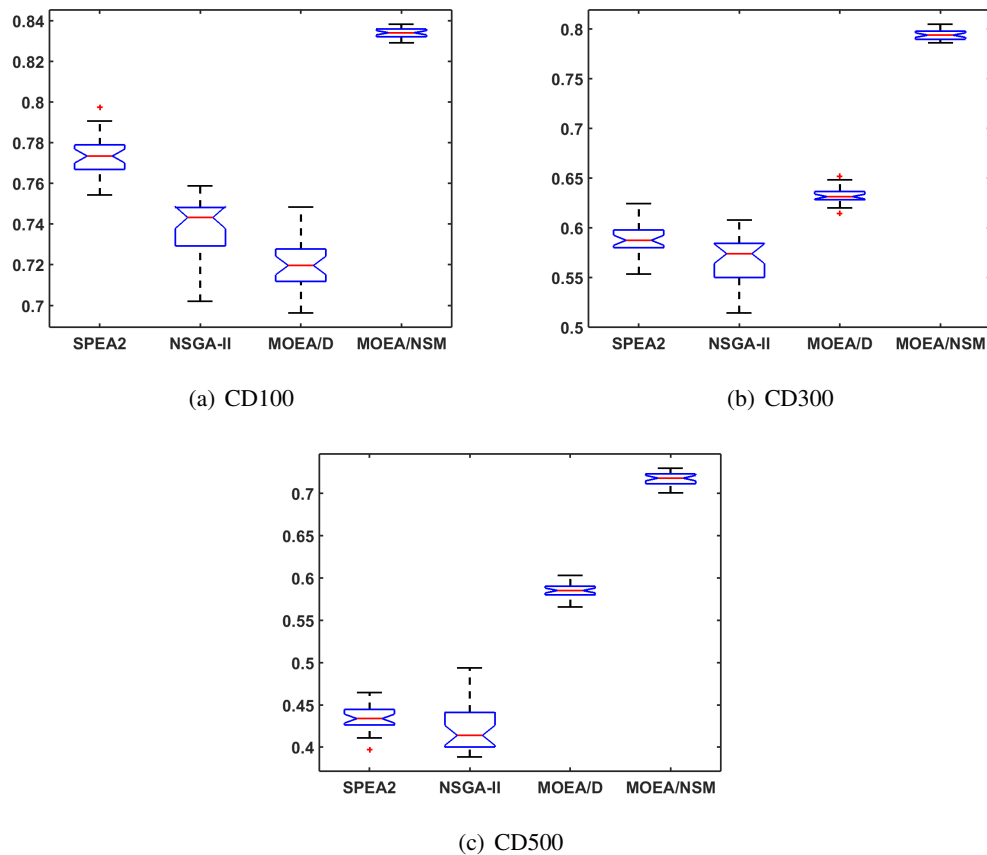


Figura 4: Valores de HV para os datasets CD100, CD300 e CD500.

bom desempenho a partir da integração das técnicas de distâncias utilizadas nas abordagens NSGA-2, SPEA2 e MOEA/D.

Agradecimentos

O autores gostariam de agradecer ao CNPq (processo: 458598/2014-3) e UTFPR pelo apoio financeiro concedido para esta pesquisa.

Referências

- Angus, D. (2007). Crowding population-based ant colony optimisation for the multi-objective travelling salesman problem. In *MCDM*, p. 333–340.
- Cornu, M., Cazenave, T., e Vanderpooten, D. (2016). Perturbed decomposition algorithm applied to the multi-objective traveling salesman problem. *Computers & Operations Research*.
- Davis, L. (1985). Applying adaptive algorithms to epistatic domains. In *IJCAI*, volume 85, p. 162–164.
- Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Wiley, New York.
- Deb, K., Agrawal, S., Pratap, A., e Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. *Lecture notes in computer science*, 1917:849–858.
- Durillo, J. J. e Nebro, A. J. (2011). jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42(10):760–771.

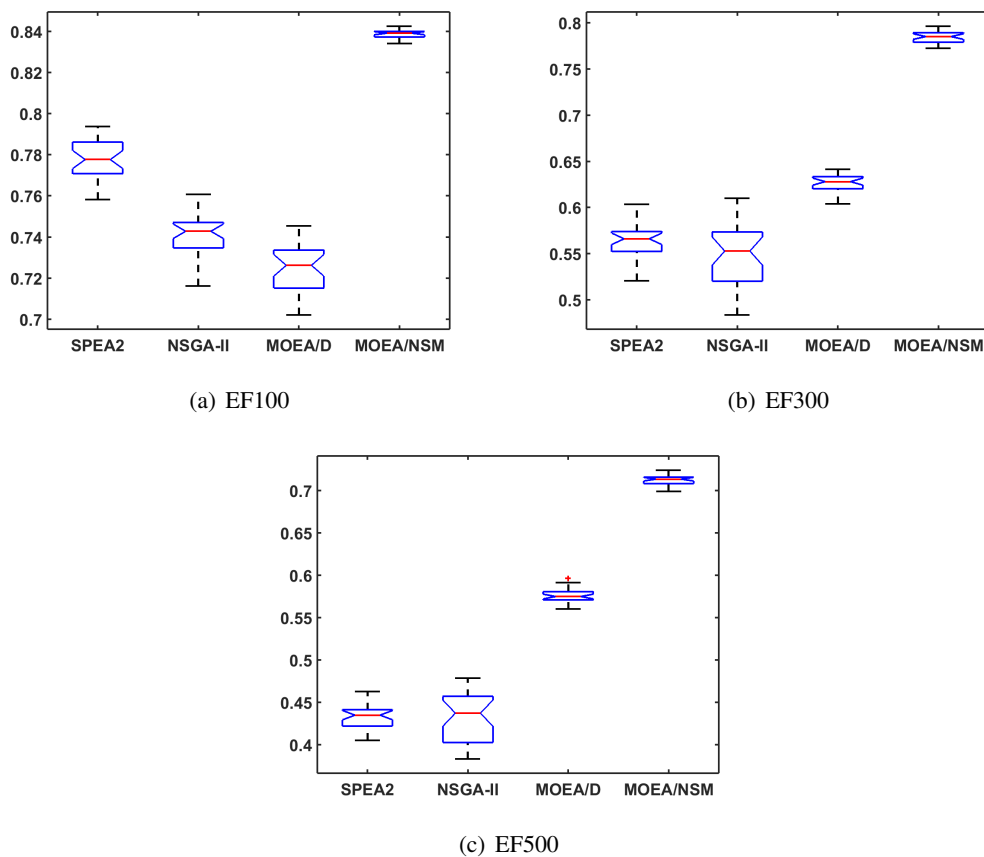


Figura 5: Valores de HV para os datasets EF100, EF300 e EF500.

Elaoud, S., Teghem, J., e Loukil, T. (2010). Multiple crossover genetic algorithm for the multiobjective traveling salesman problem. *Electronic Notes in Discrete Mathematics*, 36:939–946.

García-Martínez, C., Cordon, O., e Herrera, F. (2007). A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria tsp. *European Journal of Operational Research*, 180(1):116–148.

Gois, M. M., Sanches, D. S., Martins, J., Junior, J. B. A. L., e Delbem, A. C. B. (2013). Multi-objective evolutionary algorithm with node-depth encoding and strength pareto for service restoration in large-scale distribution systems. In *Evolutionary Multi-Criterion Optimization*, p. 771–786. Springer.

Hansen, M. P. (2000). Use of substitute scalarizing functions to guide a local search based heuristic: The case of motsp. *Journal of Heuristics*, 6(3):419–431.

Jaszkiewicz, A. (2002). Genetic local search for multi-objective combinatorial optimization. *European journal of operational research*, 137(1):50–71.

Jaszkiewicz, A. e Zielniewicz, P. (2009). Pareto memetic algorithm with path relinking for bi-objective traveling salesperson problem. *European Journal of Operational Research*, 193(3): 885–890.

Köksalan, M. e Öztürk, D. T. (2016). An evolutionary approach to generalized biobjective traveling salesperson problem. *Computers & Operations Research*.



- Krasnogor, N. e Smith, J. (2000). A memetic algorithm with self-adaptive local search: Tsp as a case study. In *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation*, p. 987–994. Morgan Kaufmann Publishers Inc.
- Kumar, R. e Singh, P. (2007). Pareto evolutionary algorithm hybridized with local search for biobjective tsp. In *Hybrid Evolutionary Algorithms*, p. 361–398. Springer.
- Lust, T. e Teghem, J. (2007). Two phase stochastic local search algorithms for the biobjective traveling salesman problem. *Proceedings of SLS-DS*, p. 21–25.
- Lust, T. e Teghem, J. (2010a). The multiobjective traveling salesman problem: a survey and a new approach. In *Advances in Multi-Objective Nature Inspired Computing*, p. 119–141. Springer.
- Lust, T. e Teghem, J. (2010b). Two-phase pareto local search for the biobjective traveling salesman problem. *Journal of Heuristics*, 16(3):475–510.
- Paquete, L. e Stützle, T. (2003). A two-phase local search for the biobjective traveling salesman problem. In *International Conference on Evolutionary Multi-Criterion Optimization*, p. 479–493. Springer.
- Paquete, L. e Stützle, T. (2009). Design and analysis of stochastic local search for the multiobjective traveling salesman problem. *Computers & operations research*, 36(9):2619–2631.
- Peng, W., Zhang, Q., e Li, H. (2009). Comparison between moea/d and nsga-ii on the multi-objective travelling salesman problem. In *Multi-objective memetic algorithms*, p. 309–324. Springer.
- Psychas, I.-D., Delimpasi, E., e Marinakis, Y. (2015). Hybrid evolutionary algorithms for the multiobjective traveling salesman problem. *Expert Systems with Applications*, 42(22):8956–8970.
- Samanlioglu, F., Ferrell, W. G., e Kurz, M. E. (2008). A memetic random-key genetic algorithm for a symmetric multi-objective traveling salesman problem. *Computers & Industrial Engineering*, 55(2):439–449.
- Sanches, D. S. (2013). *Algoritmos Evolutivos Multi-Objetivo para Reconfiguração de Redes em Sistemas de Distribuição de Energia Elétrica*. PhD thesis, Universidade de Sao Paulo.
- Zhang, Q. e Li, H. (2007). Moea/d: A multi-objective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, Accepted.
- Zitzler, E., Laumanns, M., Thiele, L., Zitzler, E., Zitzler, E., Thiele, L., e Thiele, L. (2001). Spea2: Improving the strength pareto evolutionary algorithm.