

METAHEURÍSTICA EMPÍRICA CONTÍNUA

**Francisco Carlos Rodrigues Coelho^A, Ivo Chaves da Silva Junior^B, Bruno Henriques Dias^B
e Wesley Peres^A**

^A Universidade Federal de São João del-Rei – UFSJ
Campus Santo Antônio, Praça Frei Orlando, 170, 36307-352, São João del-Rei, MG

^B Universidade Federal de Juiz de Fora – UFJF
Campus da UFJF, Plataforma do Setor de Tecnologia, 36036-330, Juiz de Fora, MG

franciscocoelho@ufsj.edu.br, ivo.junior@ufjf.edu.br, bruno.dias@ufjf.edu.br,
wesley.peres@ufsj.edu.br

RESUMO

O presente trabalho apresenta um novo algoritmo de otimização denominado Metaheurística Empírica Contínua (MEC). O que se propõe é a formulação de um mecanismo de exploração do espaço de busca de forma eficiente, com um baixo número de parâmetros para se ajustar, baixa dependência das condições iniciais e baixa dispersão dos resultados encontrados em diferentes execuções do algoritmo. A metaheurística proposta é populacional, ou seja, se baseia na interação de indivíduos entre si, explorando o espaço de soluções através de avaliações da função objetivo. Para validação da MEC, a mesma é comparada com outras quatro metaheurísticas, reconhecidamente bem sucedidas, na otimização de funções matemáticas multimodais usualmente encontradas na avaliação de algoritmos de otimização. Através das comparações, o algoritmo proposto é analisado em termos de eficiência, robustez e precisão. Os resultados obtidos mostram que o mecanismo de busca aqui proposto confronta duas das principais limitações de técnicas metaheurísticas: o alto número de parâmetros e a alta variabilidade dos resultados produzidos.

PALAVRAS CHAVE: Metaheurística Empírica Contínua. Funções Multimodais. Robustez. Metaheurísticas Populacionais. Busca Global.

Tópico: MH - Metaheurísticas

ABSTRACT

The present work presents a new optimization algorithm called Empirical Continuous Metaheuristic (ECM), where an efficient exploitation mechanism is proposed. The main features of this population-based algorithm are the low number of adjustment parameters and the low dispersion of results considering a set of different simulations. So the exploitation of the solution-space is done by iterations of the individuals among themselves, through objective function evaluations. In order to validate the ECM search mechanism, it is compared to other four well succeed metaheuristics. The comparisons are carried out through the optimization of diverse multimodal objective functions, usually found in optimization algorithms evaluations. So, the proposed metaheuristic is analyzed regarding efficiency, robustness and accuracy. The obtained results show that the ECM confronts two of the main limitations of many metaheuristics algorithm: The high number of parameters; and the high dispersion of final results.

KEYWORDS: Empirical Continuous Metaheuristic. Multimodal Functions. Robustness. Population Based Metaheuristics. Global Search.

Topic: MH - Metaheuristics

1. Introdução

Algoritmos de busca inteligente, com inspiração na natureza, vêm sendo aplicados na resolução de complexos problemas de otimização em diversas áreas do conhecimento. Recentes aplicações exemplificam a versatilidade e eficiência de técnicas metaheurísticas, como por exemplo: Algoritmo Genético (Holland, 1975) utilizado na tecnologia de reconhecimento facial (Zhi & Liu, 2019); *Particle Swarm Optimization - PSO* (Kennedy & Eberhart, 1995) aplicado ao problema de rastreamento do ponto de máxima potência em painéis fotovoltaicos (Renaudineau et al., 2015); Recozimento Simulado (Kirkpatrick et al., 1983) para mapear a expressão (efetivação da característica) de genes presentes em um indivíduo (Biswas & Acharyya, 2014); e *Bat Algorithm* (Yang, 2010b) utilizado para estimação futura do valor de mercado do cobre (Dehghani & Bogdanovic, 2017).

O grande apelo para utilização destas técnicas inteligentes é a possibilidade de resolução de problemas multimodais, com inúmeros máximos e mínimos locais, o que é ocasionado por uma região de solução não convexa, semelhante à da Figura 1.

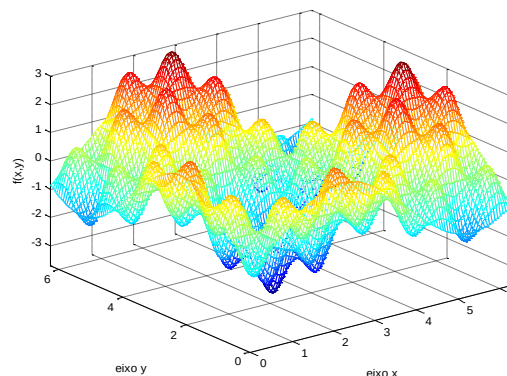


Figura 1 - Função matemática não convexa.

Os métodos clássicos de otimização, quando aplicados a problemas não convexas, tendem a convergir para ótimos locais, uma vez que são altamente dependentes das condições iniciais (Bazaraa et al., 2006). Quando se trata de metaheurísticas, estes algoritmos possuem mecanismos para se desprender dos vales/picos das regiões de solução, aumentando assim as possibilidades de encontro de soluções ótimas globais, com maior independência em relação às soluções iniciais consideradas.

Não obstante a aplicabilidade de algoritmos inteligentes, existem algumas limitações com relação a estas técnicas como, por exemplo: (i) variabilidade dos resultados obtidos; (ii) número de parâmetros para se ajustar; e (iii) tempo computacional para se atingir a solução ótima.

Tendo em vista a aplicabilidade e limitações destes algoritmos, uma metaheurística foi formulada com o intuito de confrontar estes problemas. Em suma, o que se propõe neste trabalho é uma metaheurística eficiente, robusta com relação às condições iniciais, capaz de produzir resultados com baixa dispersão e com um baixo número de parâmetros de ajuste.

2. Formulação da Metaheurística Empírica Contínua

A Metaheurística Empírica Contínua (MEC) é populacional, ou seja, possui um conjunto de indivíduos que se deslocam sobre o espaço de solução no decorrer de um processo iterativo, assim como os algoritmos do *PSO* e do *Bat Algorithm*. Seu mecanismo de busca consiste na iteração destes indivíduos entre si, e também com “passos” aleatórios. De modo geral, estas iterações ocorrem com base em avaliações sucessivas das soluções que vão surgindo ao longo da execução do algoritmo.

Tendo em vista que um dos desafios originados do uso de metaheurísticas é a quantidade de parâmetros para se ajustar, buscou-se criar um mecanismo de busca eficiente com um baixo número de parâmetros. Sendo a MEC proposta populacional e iterativa, dois destes parâmetros

são: número de indivíduos e número máximo de iterações; comuns a todas as metaheurísticas populacionais. Além destes dois, a metaheurística proposta possui apenas outros dois parâmetros, totalizando quatro.

A Figura 2 mostra o fluxograma básico da MEC. No primeiro passo (**P1**), a população é inicializada de forma aleatória, por uma distribuição uniforme de probabilidade, delimitada no espaço de solução. Neste momento, os quatro parâmetros do algoritmo são definidos, sendo N o número de indivíduos, ITE o número máximo de iterações, G o número de grupos em que a população será dividida e $STAG$ o número de iterações necessário para se considerar que o algoritmo se encontra estagnado. Cada indivíduo se localiza em uma posição $\mathbf{x}_{n,g}^t$ de índice n , na iteração t e pertence a algum grupo g . Deste modo, $\mathbf{x}_{n,g}^t$ é um vetor referente a uma possível solução para o problema em análise.

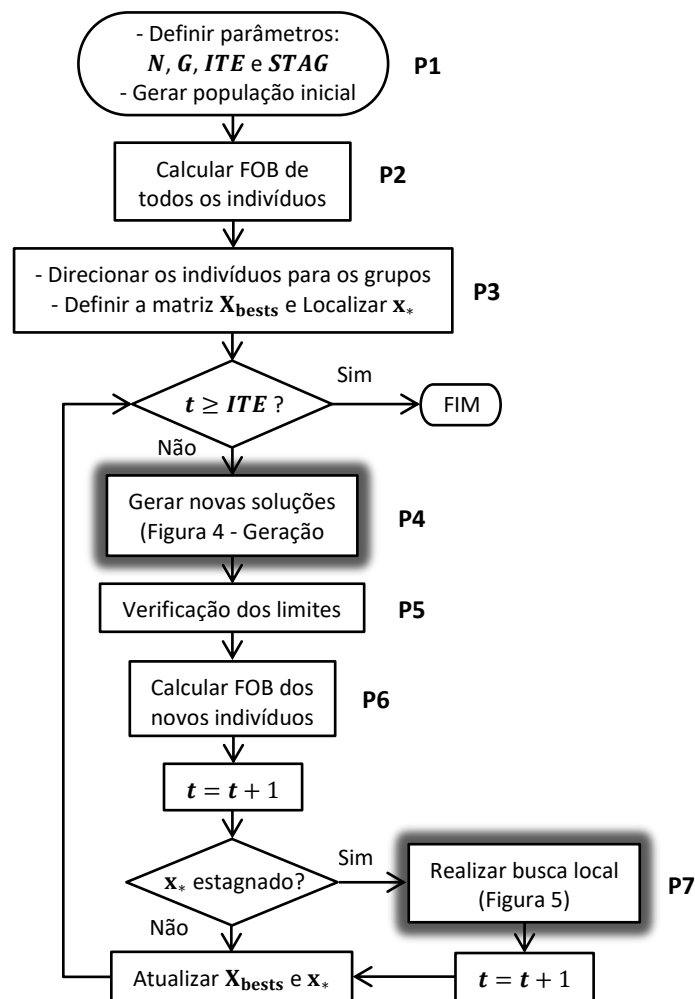


Figura 2 - Fluxograma básico da Metaheurística Empírica Contínua (MEC).

Em **P2** as soluções são avaliadas através da função objetivo (FOB) e classificadas em uma lista decrescente de aptidão. Em **P3**, a população é dividida em grupos. A alocação de cada indivíduo em determinado grupo é realizada de acordo com sua aptidão, de modo que cada grupo possua um conjunto diversificado de soluções. O indivíduo melhor classificado é direcionado para o Grupo 1, o segundo melhor, para o Grupo 2 e assim sucessivamente. Cada rodada de distribuição de indivíduos/soluções dura o número de grupos, isto é, se o número de grupos for quatro, o quinto indivíduo da lista será alocado no “Grupo 1” e o processo se repete. A Figura 3 ilustra esta dinâmica para o caso de seis indivíduos e três grupos.

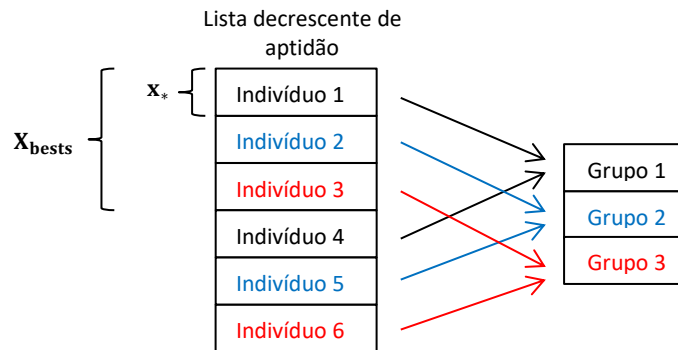


Figura 3 - Esquema de divisão da população inicial em grupos.

Após esta divisão, define-se a matriz x_{bests} contendo o melhor indivíduo de cada grupo, e o melhor indivíduo entre todos x_* é localizado.

Passadas estas primeiras etapas, o algoritmo entra em seu processo iterativo. Não sendo a última iteração, parte-se para a etapa **P4**, na qual novas soluções são propostas. A Figura 4 detalha este passo através de um fluxograma.

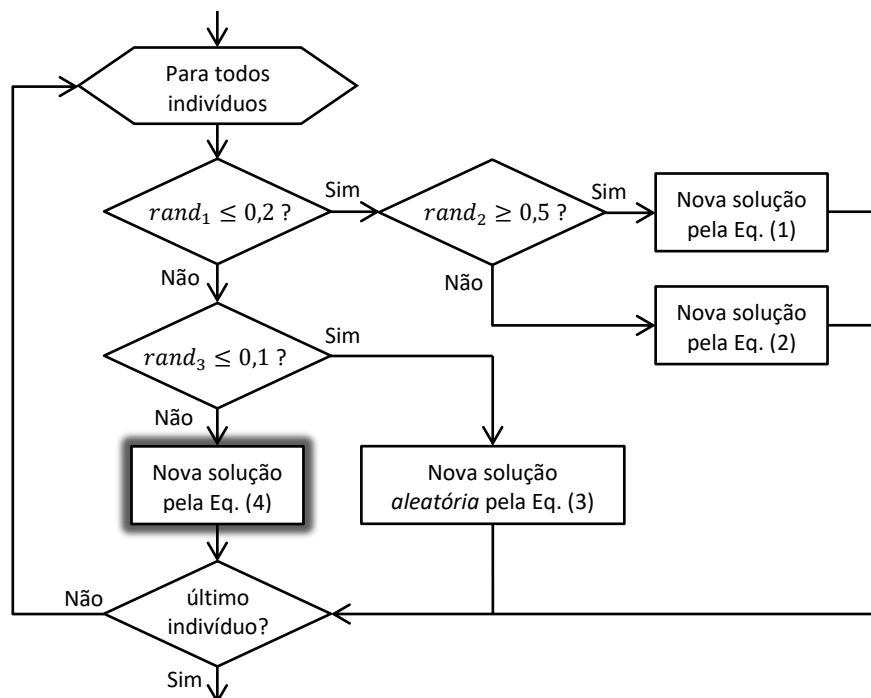


Figura 4 - Geração de novas soluções (**P4**).

A proposição de novas soluções pode ser compreendida como sendo o deslocamento dos indivíduos sobre o espaço de solução, e a maneira como este deslocamento se processa depende de algumas decisões aleatórias baseadas em probabilidade. A variável $rand \in [0; 1]$ corresponde a um valor aleatório proveniente de uma distribuição uniforme, gerado a cada decisão. Portanto, logo no início do processo, há uma probabilidade de 20% que as novas soluções sejam geradas pelas Equações (1) ou (2), que estão definidas a seguir.

$$x_{n,g}^{t+1} = x_{n,g}^t + E \quad (1)$$

$$x_{n,g}^{t+1} = x_{n,g}^t + rand * (x_* - x_{n,g}^t) \quad (2)$$

Caso a nova solução seja criada através da Equação (1), a movimentação do indivíduo será completamente aleatória sobre o espaço de solução, pois E corresponde a um vetor de

natureza aleatória, que se soma a atual posição do indivíduo. Se por outro lado, a Equação (2) determinar o deslocamento do indivíduo, o mesmo dará um “passo” direto no sentido do melhor indivíduo da população $\mathbf{x}_*$ na presente iteração.

Não ocorrendo o deslocamento pelas opções anteriores, existe 10% de chances de a nova solução criada ser completamente nova, isto é, descarta-se a posição atual e o indivíduo recebe um novo vetor solução, aleatório, de acordo com a Equação (3).

$$\mathbf{x}_{n,g}^{t+1} = \mathbf{x}^{New} \quad (3)$$

Como se nota no fluxograma da Figura 4, a probabilidade das novas soluções serem criadas pelas equações anteriores correspondem ao caminho de menor probabilidade. Assim sendo, a Equação (4) representa o principal mecanismo de busca presente no algoritmo proposto (passo destacado na Figura 4).

$$\mathbf{x}_{n,g}^{t+1} = \mathbf{x}_{n,g}^t + rand * (\mathbf{X}_{bests}(g) - \mathbf{x}_{n,g}^t) + E \quad (4)$$

Explora-se o espaço de busca com base na atual posição do indivíduo, que será acrescida de um deslocamento na direção do melhor indivíduo de seu grupo, mais um passo aleatório.

Ressalta-se aqui, a diferença entre o algoritmo proposto e técnicas semelhantes como *PSO* e *Bat Algorithm*. De maneira distinta dos métodos citados, os indivíduos da MEC não possuem um vetor de “velocidade” para seu deslocamento, o que possibilita mudanças bruscas na direção, acelerando o processo de busca. Além disto, no algoritmo *PSO* os indivíduos possuem outra componente de deslocamento, que é referente à melhor posição ocupada pelo indivíduo até então. Em se tratando do *Bat Algorithm*, as soluções resultantes da etapa de deslocamento não são automaticamente aceitas como na MEC. Esta decisão só é tomada se o deslocamento resultar em uma solução melhor do que a anterior. Por fim, não há divisão em grupos nestes algoritmos.

Em **P5**, os limites do espaço de busca são verificados. Caso as novas soluções propostas tenham excedido estes limites, as mesmas são recuadas para os referidos pontos limítrofes. Em **P6**, uma nova lista de aptidões é criada com os resultados das novas avaliações dos indivíduos.

Neste instante, toma-se a decisão de realizar, ou não, busca local. Trata-se do sétimo passo na Figura 2 (**P7**, em destaque). Esta decisão é baseada na estagnação da melhor solução $\mathbf{x}_*$. Caso não haja melhora em um determinado número de iterações (**STAG**), o algoritmo entra em sua fase de intensificação, exibida no fluxograma da Figura 5, tendo em vista um problema de minimização.

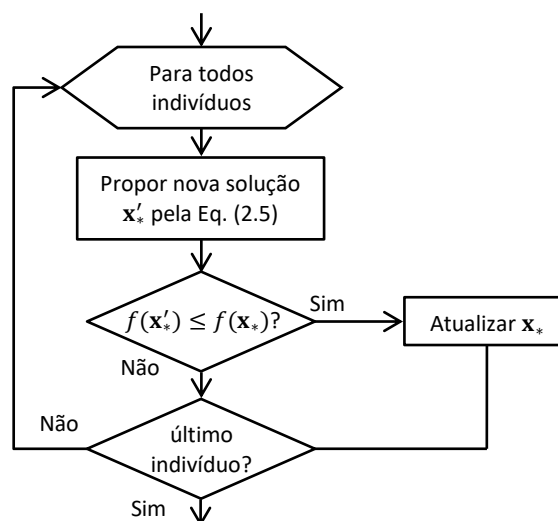


Figura 5 - Etapa de busca local (P7).

A busca local é realizada N vezes por meio da Equação (5), que consiste em uma pequena perturbação nas proximidades da melhor solução $\mathbf{x}_*$.

$$\mathbf{x}'_* = \mathbf{x}_* + \boldsymbol{\epsilon} \quad (5)$$

Onde $\boldsymbol{\epsilon}$ é um vetor aleatório de pequeno módulo. Após este pequeno deslocamento, a solução produzida é comparada com a anterior. Caso a nova seja melhor, o indivíduo $\mathbf{x}_*$ é atualizado. Esta característica confere um caráter dinâmico à busca local, pois este indivíduo pode ser atualizado múltiplas vezes, aumentando assim as chances de se atingir o mínimo local daquela região da função objetivo. Finalizada esta etapa, outra iteração é contabilizada, pois esforço computacional foi requerido para realização desta tarefa. Neste momento, a função objetivo foi calculada N vezes.

Para concluir o processo, $\mathbf{X}_{\text{bests}}$ e $\mathbf{x}_*$ são atualizados e o critério de parada é verificado novamente. Sendo a última iteração, o algoritmo proposto na MEC é finalizado.

3. Validação da Metaheurística Empírica Contínua

Para validação da MEC, foram realizadas comparações da mesma com outras metaheurísticas, análises acerca de sua robustez com relação à população inicial e com relação à função objetivo que se deseja otimizar. Aplicou-se o algoritmo na minimização de funções matemáticas não convexas, usualmente encontradas na avaliação de diferentes algoritmos de otimização multimodais. Todas estas funções possuem seus ótimos globais conhecidos.

Além de comparações com o *Bat Algorithm* (BA) e *PSO*, o desempenho da metaheurística proposta em sua versão contínua foi comparado com a metaheurística *Cuckoo Search* (CS) (Yang & Deb, 2009) e *Firefly Algorithm* (FA) (Yang, 2009), técnicas reconhecidamente bem sucedidas na otimização de diversos problemas de engenharia. A Tabela 1 mostra todos os parâmetros utilizados nas simulações.

Tabela 1 - Parâmetros utilizados nas metaheurísticas comparadas.

Metaheurística	Parâmetro	Valor	Descrição
MEC	N	40	Número de indivíduos
	ITE	40	Número de iterações
	G	4	Número de grupos
	$STAG$	7	Parâmetro de estagnação
BA	n	40	Número de indivíduos
	ite	40	Número de iterações
	α	0,9	Constante para decréscimo do volume
	γ	0,8	Constante para acréscimo da taxa de emissão
	r^0	1	Taxa máxima de emissão de pulso
	f_{min}	0	Frequência mínima
	f_{max}	0,15	Frequência máxima
CS	n	40	Número de indivíduos
	ite	40	Número de iterações
	α	1	Peso do passo <i>Lévy Flyght</i>
	λ	2	Expoente da distribuição de <i>Lévy</i>
	P_a	0,25	Probabilidade de abandono do ninho
FA	n	40	Número de indivíduos
	ite	40	Número de iterações
	α	0,1	Peso do passo aleatório
	γ	0,4	Coeficiente de absorção de luz
	β_0	0,1	Atratividade máxima
	δ	0,95	Fator de redução do passo aleatório
PSO	n	40	Número de indivíduos
	ite	40	Número de iterações
	c_1	0,3	Peso da informação local
	c_2	0,7	Peso da informação global
	θ	1	Fator de inércia sobre a velocidade

Todas as metaheurísticas contaram com um número fixo de 40 indivíduos explorando o espaço de busca ao longo de 40 iterações. Os demais parâmetros presentes na Tabela 1 são provenientes de faixas recomendadas pelos próprios autores destes algoritmos. O “ajuste fino”, ou *tuning*, destes parâmetros foi feito com base em análise de sensibilidade destes com relação aos valores das funções matemáticas, visando à obtenção dos melhores resultados possíveis para todas as metaheurísticas comparadas. Ressalta-se que a metaheurística proposta é a que possui o menor número de parâmetros para se ajustar como se nota na Figura 6.

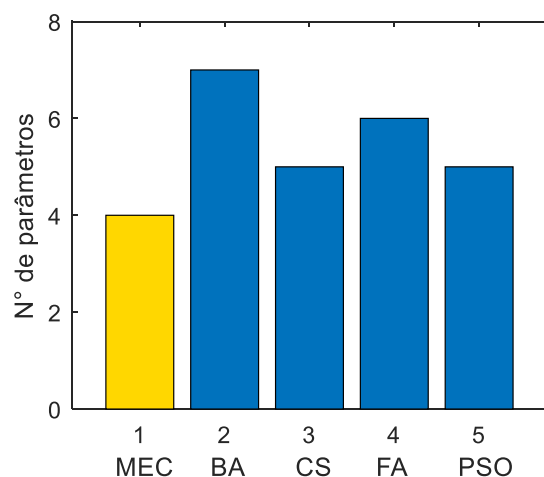


Figura 6 - Número de parâmetros por metaheurística.

Ressalta-se, neste momento, a importância da escolha de um critério de parada adequado, algo necessário em qualquer processo iterativo, como é o caso dos algoritmos metaheurísticos. Conforme descrito em (Yang, 2010b), existem duas maneiras básicas de se realizar comparações de desempenho entre metaheurísticas. Uma delas é através de um número fixo de avaliações da função objetivo, como por exemplo, 40 indivíduos e 40 iterações, totalizando 1600 avaliações. Outra opção é mensurar o número necessário de avaliações da FOB para chegar ao ponto ótimo com certa tolerância.

Neste trabalho, optou-se pela primeira opção com a intenção de se realizar uma comparação equânime. Em suma, todos os algoritmos contam com um número fixo de avaliações da FOB, sendo o número máximo de iterações o critério de parada.

A seguir, são apresentados os resultados das comparações realizadas para validação da MEC. Uma das quinze funções matemáticas será analisada em maiores detalhes. Trata-se da função *Eggcrate*, presente em (Yang, 2010b). As demais funções multimodais utilizadas se encontram em (Ali et al., 2005).

Toda a implementação computacional das metaheurísticas aqui apresentadas foram conduzidas no ambiente do *software* MATLAB, versão 2010a, com *hardware* Intel® Core™ i5, 2.80GHz e 4GB de memória RAM.

3.1 Função *Eggcrate*

A formulação matemática da função *Eggcrate* se encontra nas equações (6) e (7), com a última explicitando os limites nos quais estão compreendidas as variáveis independentes x e y .

$$f(x, y) = x^2 + y^2 + 25(\sin^2 x + \sin^2 y) \quad (6)$$

$$-2\pi \leq x, y \leq 2\pi \quad (7)$$

O que se deseja encontrar no processo de busca é o ponto mínimo da função *Eggcrate*, de coordenadas $f(0, 0) = 0$. A Figura 7 exhibe o aspecto tridimensional desta função, com as regiões

em vermelho significando valores altos, portanto indesejáveis, e as regiões em azul representando os pontos de baixo valor da função objetivo.

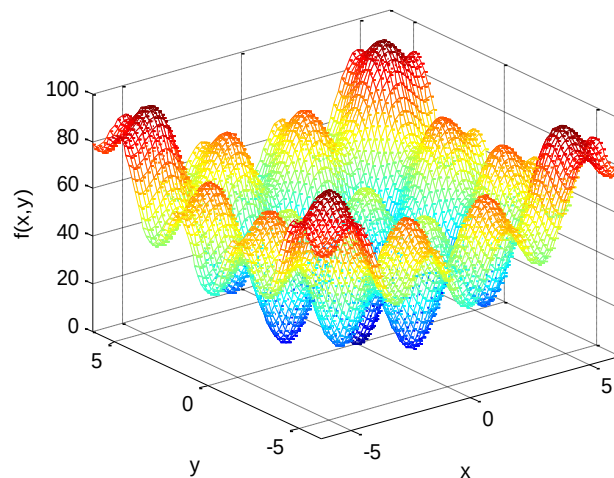


Figura 7 - Função *Eggcrate*.

A. Comparação entre metaheurísticas

Partindo da mesma população inicial, cada algoritmo foi executado mil vezes. Desta maneira, é possível realizar uma análise estatística sobre o desempenho dos mesmos. Considerando o total de simulações, as mil soluções finais obtidas pelas metaheurísticas foram armazenadas em cinco conjuntos. A Tabela 2 contém alguns dos dados estatísticos destes conjuntos.

Tabela 2 - Dados estatísticos da função *Eggcrate*.

	MEC	BA	CS	FA	PSO
Média	1,88e-5	0,1861	0,0123	0,0243	0,1229
Desvio padrão	2,22e-5	0,2251	0,0104	0,0205	0,1095
Mínimo	7,30e-9	8,07e-5	1,73e-6	2,29e-7	8,51e-5
Encontro do ótimo global	99,8%	40,3%	99,5%	92,6%	44,5%

Os dados estatísticos presentes na Tabela 2 mostram que a MEC obteve, em geral, melhores resultados com relação aos outros métodos. Dentre os cinco conjuntos de dados finais, a metaheurística proposta apresentou a média mais baixa, o menor desvio padrão, o mínimo valor encontrado e a mais alta taxa de encontro do ótimo global. O algoritmo *Cuckoo Search* obteve uma taxa de encontro do ótimo global de 99,5%, valor bastante próximo dos 99,8% da MEC.

Esta última informação contida na Tabela 2, chamada de “Encontro do ótimo global”, representa a porcentagem de vezes, nas mil simulações, em que os métodos atingiram a região considerada como ótimo global. Trata-se de uma pequena distância euclidiana do ponto ótimo. A região considerada como sendo ótimo global da função *Eggcrate* está definida na Equação (8).

$$-0,05 \leq x, y \leq 0,05 \quad (8)$$

Além da Tabela 2, o conjunto de dados pode ser analisado sob uma perspectiva gráfica, através de um diagrama de caixa, frequentemente referido como *box plot*. Esta ferramenta estatística é bastante útil quando se deseja realizar comparações entre dados, pois descreve, simultaneamente, diversas características do conjunto de dados como: centro; dispersão; desvio de simetria; e discrepâncias com relação ao conjunto (Montgomery & Runger, 2014).

A Figura 8 mostra um *box plot*, com os dados obtidos ao final das simulações. A interpretação das informações presentes no *box plot* é relativamente simples. A “caixa” contém 50% de todos os valores presentes nos dados, a linha vermelha, que a divide, representa a

mediana do conjunto. As linhas tracejadas que se estendem além da caixa são chamadas de “caudas”. Estes prolongamentos se encerram com o fim dos dados. Os valores discrepantes, ou *outliers*, são sinalizados por cruzeiros vermelhos. Destaca-se que para efeito de análise estatística, estes valores não compõem o conjunto principal de dados (Montgomery & Runger, 2014).

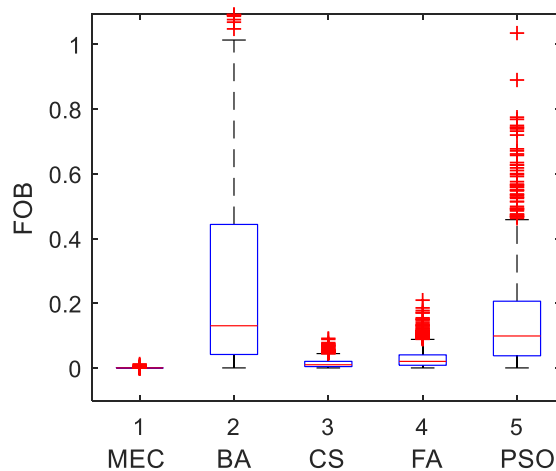


Figura 8 - Box plot com os dados da função *Eggcrate*.

Como se nota na Figura 8, a MEC produziu resultados com a mais baixa dispersão dentre os algoritmos comparados. Seus resultados praticamente não diferenciam o início e o fim do *box plot*, dada a alta precisão apresentada pela metaheurística proposta na otimização da função *Eggcrate*. Dentre os algoritmos comparados, BA e PSO produziram os resultados mais dispersos, com alto número de *outliers* presentes nos dados do segundo, enquanto a MEC apresenta o menor número destes valores discrepantes.

B. Robustez com relação à população inicial

Além da capacidade de encontro de boas soluções, é desejável em uma metaheurística que a mesma possua relativa independência com relação à suas condições iniciais. Tal característica pode ser compreendida como robustez em relação à população inicial, pela qual, o mecanismo de busca inicia seu processo de varredura do espaço de soluções (Vasant et al., 2016).

Para avaliação deste importante aspecto, realizaram-se 1000 repetições da MEC aplicada a minimização da função *Eggcrate* em duas situações. Na primeira, a MEC partiu de um mesmo conjunto de soluções iniciais. Na segunda situação, a execução repetida da MEC foi realizada partindo de soluções iniciais diferentes e aleatórias, provenientes de uma distribuição uniforme de probabilidade.

A Figura 9 exibe o *box plot* da MEC nas duas situações, sendo as siglas PIF – População Inicial Fixa e PIV – População Inicial Variável. Com o intuito de verificar até que ponto se pode afirmar que as duas amostras presentes na Figura 9 são estatisticamente iguais com rigor, realizou-se um teste de hipóteses.

O *Teste t para Duas Amostras*, descrito em (Montgomery & Runger, 2014), permite que se verifique a probabilidade de que as diferenças presentes em duas amostras **não** sejam devidas ao acaso, trata-se do *valor-p*. É usual o nível de significância de 5%, que significa que se o *valor-p* se situar abaixo de 5%, rejeita-se a *hipótese nula*, que estabelece a igualdade entre as amostras.

Considerando então este nível de significância, ou *intervalo de confiança* de 95%, aplicou-se o referido teste às duas amostras obtidas. O *valor-p* encontrado ficou próximo a 45%, não sendo possível, portanto, a rejeição da *hipótese nula*. O resultado proveniente do *Teste t para Duas Amostras* é de que não se pode afirmar que as duas amostras possuem diferenças significativas, o que é um indício da robustez da metaheurística proposta com relação à população inicial.

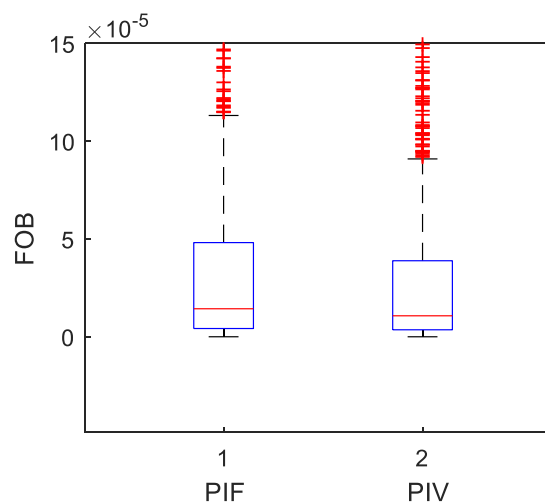


Figura 9 - Robustez da MEC com relação à população inicial (função *Eggcrate*).

3.1 Demais funções

Outra característica desejável em uma metaheurística é o bom desempenho quando aplicada a uma gama de diversos problemas de otimização. Ou seja, espera-se que a mesma seja robusta com relação à função objetivo que se deseja otimizar.

Para avaliar a MEC neste quesito, realizaram-se simulações com outras funções matemáticas. De modo semelhante às avaliações da função *Eggcrate*, foram conduzidas 1000 repetições com cada uma das cinco metaheurísticas, sempre partindo das mesmas soluções iniciais. Dadas as características de *hardware* mencionadas na seção anterior, cada conjunto de execuções de um mesmo algoritmo dura cerca de 10 segundos.

A Tabela 3 mostra o índice percentual de encontro do ótimo global atingido pelos algoritmos de otimização, para cada função, com a respectiva média de sucesso ao final. Além das funções matemáticas, as coordenadas dos respectivos ótimos globais e os limites das variáveis independentes se encontram em (Ali et al., 2005).

Tabela 3 - Índice de encontro do ótimo global: Várias funções.

Função	MEC	BA	CS	FA	PSO
<i>Ackley</i>	100%	52,3%	100%	99,1%	43,0%
<i>Aluffi-Pentini</i>	100%	41,4%	75,1%	59,2%	41,6%
<i>Becker and Lago*</i>	99,8%	24,9%	99,9%	36,1%	23,0%
<i>Bohachevsky 1</i>	100%	37,1%	100%	77,3%	38,9%
<i>Bohachevsky 2</i>	99,1%	15,9%	100%	55,7%	35,0%
<i>Camel Back 3</i>	99,9%	55,6%	100%	95,5%	65,1%
<i>Cosine Mixture</i>	100%	54,0%	90,4%	85,7%	67,2%
<i>Easom</i>	99,9%	24,3%	73,5%	60,5%	16,8%
<i>Eggcrate</i>	99,8%	40,3%	99,5%	92,6%	44,5%
<i>Exponential</i>	100%	69,1%	89,0%	83,8%	94,0%
<i>Michalewicz</i>	99,8%	33,8%	85,7%	58,9%	30,2%
<i>Multi-Gaussian</i>	70,6%	18,4%	39,0%	45,9%	25,3%
<i>Rastrigin</i>	60,7%	49,3%	74,2%	62,8%	58,1%
<i>Rosenbrock</i>	75,6%	42,5%	73,5%	27,1%	72,9%
<i>Schwefel</i>	60,8%	14,7%	75,8%	8,4%	18,3%
Média total	91,1%	38,3%	85,0%	63,3%	44,9%

* Múltiplos ótimos globais

Observa-se na Tabela 3, que os maiores índices de alcance do ponto ótimo foram obtidos pela MEC para a maioria das funções. Em cinco oportunidades, a metaheurística proposta chegou à região ótima em todas as 1000 simulações. A média total mostra que o algoritmo proposto se mostrou competitivo em relação à metaheurísticas bem sucedidas existentes na literatura especializada, alcançando o ótimo global em 91,1% das oportunidades.

A Figura 10 apresenta, graficamente, os dados com as taxas de encontro do ótimo global, provenientes da Tabela 3, por parte de cada uma das metaheurísticas em análise.

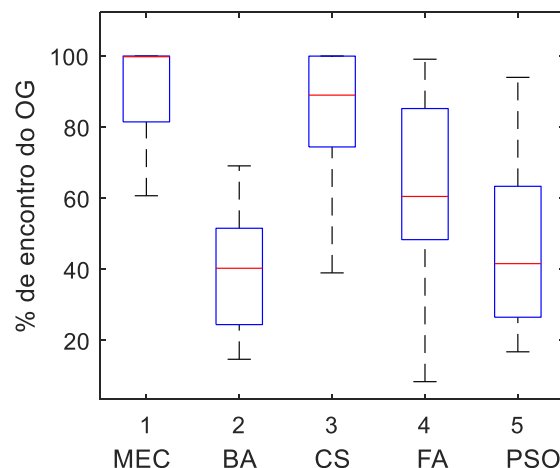


Figura 10 - Índice de encontro do ótimo global sobre várias funções.

Como pode se observar na Figura 10, os dados da MEC se encontram relativamente menos dispersos, com obtenção mínima de ótimo global em 60%, e máxima em 100%. Adicionalmente, a mediana resultante da metaheurística proposta se encontra praticamente no valor de 100%. O *Firefly Algorithm* por sua vez, apresentou melhores resultados em relação ao BA e PSO, porém com uma dispersão mais alta.

Em suma, os resultados apresentados permitem a afirmação de que para estas funções, o algoritmo proposto apresentou maior robustez, além de um maior índice de sucesso no que diz respeito à otimalidade global.

É oportuno mencionar o artigo intitulado “*No free lunch theorems for optimization*” (Wolpert & Macready, 1997), em que os autores demonstram que sempre haverá problemas nos quais determinadas técnicas de otimização se sobressaem com relação a outras. Ou seja, não existe um algoritmo absolutamente superior. O desdobramento destas conclusões é o desenvolvimento de técnicas para resolução de problemas específicos (X. Yang, 2010a).

4. Conclusões

Técnicas metaheurísticas de otimização vem sendo utilizadas nas mais variadas aplicações em otimização. No entanto, estes métodos possuem algumas limitações inerentes ao processo de busca realizado. A Metaheurística Empírica Contínua (MEC) abordou de forma direta duas destas limitações: O número de parâmetros para se ajustar; e a variabilidade dos resultados finais em diferentes execuções.

Além do fato de que a metaheurística proposta possui apenas quatro parâmetros de ajuste, as simulações repetitivas mostraram o mais baixo nível de dispersão entre as metaheurísticas comparadas, sendo estas reconhecidamente bem sucedidas na literatura especializada (*Bat Algorithm*, *Cuckoo Search*, *Firefly Algorithm* e *Particle Swarm Optimization*).

A eficiência da MEC ficou comprovada para quinze funções matemáticas multimodais usualmente encontradas na avaliação de algoritmos de otimização. Também foram feitas análises a cerca da robustez da metaheurística proposta com relação às condições iniciais, com indícios de que a mesma possui uma alta capacidade de busca global. Acredita-se que o bom desempenho da

MEC se deva ao bom equilíbrio entre diversificação (busca global) e intensificação (busca local). Tal equilíbrio esta presente no mecanismo proposto para exploração do espaço de soluções.

Apesar dos resultados apresentados serem promissores, é necessário que a metaheurística proposta seja avaliada em aplicações reais, para que se possa ter uma noção mais apurada do seu alcance como ferramenta de otimização.

Referências

- Ali, M. M., Khompatraporn, C., & Zabinsky, Z. B. (2005). A Numerical Evaluation of Several Stochastic Algorithms on Selected Continuous Global Optimization Test Problems. *Journal of Global Optimization*, 31(4), 635–672. <https://doi.org/10.1007/s10898-004-9972-2>
- Bazaraa, M., Sherali, H., & Shetty, C. (2006). *Nonlinear Programming Theory and Algorithms*. <https://doi.org/10.1002/0471787779>
- Biswas, S., & Acharyya, S. (2014). Gene expression profiling by estimating parameters of gene regulatory network using simulated annealing: A comparative study. In *2014 IEEE International Advance Computing Conference (IACC)* (pp. 56–61). IEEE. <https://doi.org/10.1109/IAAdCC.2014.6779294>
- Dehghani, H., & Bogdanovic, D. (2017). Copper price estimation using bat algorithm. *Resources Policy*, (October), 1–7. <https://doi.org/10.1016/j.resourpol.2017.10.015>
- Douglas C. Montgomery; George C. Runger. (2014). *Applied Statistics and Probability for Engineers* (Sixth Edit). John Wiley & Sons, Inc.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. (M. Press, Ed.) (1st ed.). London, England.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks* (Vol. 4, pp. 1942–1948). IEEE. <https://doi.org/10.1109/ICNN.1995.488968>
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220(4598), 671–680. <https://doi.org/10.1126/science.220.4598.671>
- Renaudineau, H., Donatantonio, F., Fontchastagner, J., Petrone, G., Spagnuolo, G., Martin, J. P., & Pierfederici, S. (2015). A PSO-based global MPPT technique for distributed PV power generation. *IEEE Transactions on Industrial Electronics*, 62(2), 1047–1058. <https://doi.org/10.1109/TIE.2014.2336600>
- Vasant, P. M., Rahman, I., Singh Mahinder Singh, B., & Abdullah-Al-Wadud, M. (2016). Optimal power allocation scheme for plug-in hybrid electric vehicles using swarm intelligence techniques. *Cogent Engineering*, 3(1), 1–26. <https://doi.org/10.1080/23311916.2016.1203083>
- Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67–82. <https://doi.org/10.1109/4235.585893>
- Yang, X.-S. (2009). Firefly Algorithms for Multimodal Optimization. In *Proceedings of the 5th International Conference on Stochastic Algorithms: Foundations and Applications* (pp. 169–178). https://doi.org/10.1007/978-3-642-04944-6_14
- Yang, X.-S. (2010). A New Metaheuristic Bat-Inspired Algorithm. In *Studies in Computational Intelligence* (Vol. 284, pp. 65–74). https://doi.org/10.1007/978-3-642-12538-6_6
- Yang, X.-S., & Suash Deb. (2009). Cuckoo Search via Lévy Flights. In *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)* (pp. 210–214). IEEE. <https://doi.org/10.1109/NABIC.2009.5393690>
- Yang, X. (2010). *Nature-Inspired Metaheuristic Algorithms Nature-Inspired Metaheuristic Algorithms Second Edition*.
- Zhi, H., & Liu, S. (2019). Face recognition based on genetic algorithm. *Journal of Visual Communication and Image Representation*, 58, 495–502. <https://doi.org/10.1016/j.jvcir.2018.12.012>