Assimilação Gaussiana para *Clustering Search* com Aplicação em Otimização Contínua Irrestrita

André F. S. Oliveira
Universidade Federal do Maranhão (UFMA)
Av Portugueses, 1966, São Luís MA
jacobi.and@hotmail.com

Marcelo B. do Nascimento
Universidade Federal do Maranhão (UFMA)
Av Portugueses, 1966, São Luís MA
marcelognesp@gmail.com

Areolino Almeida Neto
Departamento de Informática, Universidade Federal do Maranhão (UFMA)
Av Portugueses, 1966, São Luís MA
areolino@ufma.br

Paulo R. A. Ribeiro
Curso de Engenharia da Computação, Universidade Federal do Maranhão (UFMA)
Av Portugueses, 1966, São Luís MA
paulorogeriocp@gmail.com

Alexandre C. M. Oliveira
Departamento de Informática, Universidade Federal do Maranhão (UFMA)
Av Portugueses, 1966, São Luís MA
alexandre.cesar@ufma.br

RESUMO

Este trabalho propõe novos métodos de assimilação para Busca Guiada por Agrupamentos (*Clustering Search*) específicos para aplicações relacionadas à otimização contínua. Para validação, foi desenvolvida uma aplicação baseada em ECS (*Evolutionary Clustering Search*) que é uma versão híbrida de um algoritmo genético, empregando agrupamento iterativo para identificar regiões de busca promissoras e busca local direta baseada no método de *Hooke-Jeeves*. Os novos métodos de assimilação propostos, *centro de cluster* e *centro de massa*, ambos baseados em distribuição normal, foram incorporadas a duas diferentes versões do ECS, as quais foram aplicadas a um *benchmark* de funções-teste e os resultados comparados a propostas anteriores: assimilação simples e por recombinação. Os resultados encontrados são satisfatórios e em vários casos apresentaram-se significativamente melhores, em termos estatísticos, isto é $p \leq 0,05$, ou equivalentes aos métodos já utilizados na literatura. O centro de *cluster* foi o método que obteve mais resultados estatisticamente significantes.

PALAVRAS CHAVE. Metaheurísticas, Assimilação, Busca Guiada por Agrupamento, Otimização Contínua.

ABSTRACT

This paper proposes new assimilation methods for Clustering Search for applications related to continuous optimization. In order to validate them, an application based on Evolutionary Clustering Search, which is a hybrid version of a genetic algorithm, was developed using iterative clustering to identify promising search areas and then apply local search operators in these areas. The proposed assimilation methods are called cluster center and center of mass, both of them based on normal distribution, and they were compared to previous proposals, namely simple and recombination assimilations, using a benchmark that is widely used in the literature. The results are satisfactory and in several cases they are statistically significant better, i.e. $p \leq 0,05$, or equivalent to methods already used in the literature. The cluster center method obtained the majority of results that were statistically significant.

KEYWORDS. Metaheuristics, Assimilation, Clustering Search, Continuous Optimization.

1. Introdução

Na literatura existem diversos algoritmos para a resolução de problemas de otimização contínua, tipicamente modelados usando-se variáveis de decisão com domínio definido em $\mathbb{R}$. O algoritmo *Covariance Matrix Adaptation with Evolution Strategy* (CMA-ES) tem participado de competições em eventos internacionais com significativo sucesso [Hansen et al., 2003]. Outro algoritmo conhecido e com bons resultados na resolução desse tipo de problema é o *Evolutionary Clustering Search* (ECS) [Oliveira e Lorena, 2004] que utiliza um algoritmo genético (AG) [Goldberg, 1989] associado a um otimizador local do tipo Busca Direta de Hooke-Jeeves [Hooke e Jeeves, 1961].

A Busca Evolutiva guiada por Agrupamento (*Evolutionary Clustering Search – ECS*) usa agrupamento de soluções candidatas, dividindo o espaço de busca total em pequenos subespaços onde se concentram soluções candidatas de qualidade. Podem-se utilizar algoritmos de busca local para fazer varreduras dentro dos agrupamentos mais promissores, com ganhos de desempenho e redução de custo computacional [Oliveira e Lorena, 2004]. Apesar de existirem várias contribuições baseadas em *Clustering Search* em otimização combinatória [Oliveira et al., 2013], poucos trabalhos têm sido devotados à proposição de novos operadores específicos para otimização contínua.

Neste trabalho, propõe-se utilizar distribuição normal como estratégia de atualização dos centros dos agrupamentos, tendo em vista que cada agrupamento modela um subespaço de busca através de uma solução de referência, chamada de centro, a partir da qual, podem ser obtidas novas soluções dentro de um certo raio, com distribuição uniforme.

Até o momento, não existem na literatura, operadores que promovam a atualização de agrupamentos assumindo que a região promissora deve ter uma distribuição gaussiana em torno do centro. Para validar a proposta, novas versões do ECS, incorporando-os, são apresentadas e comparadas às anteriores.

A organização deste trabalho é feita de forma em que na primeira seção discute-se sobre algoritmos evolutivos e explica-se seus mecanismos. Na seção 4 discorre-se a respeito da busca guiada por agrupamentos, onde ela é formalizada e tem seus processos e componentes explicados, na seção 5 é feita a explicação dos novos métodos de assimilação que são propostos neste trabalho. Por último, na seção 6 tem-se a análise dos resultados computacionais, análise estatística desses resultados e a conclusão do trabalho na seção 7.

2. Algoritmos Evolutivos

Algoritmos evolutivos utilizam-se de modelos computacionais baseados na evolução natural para resolver problemas. Todas as abordagens evolutivas têm em comum o conceito de simulação da evolução das espécies, que se utilizam dos processos de seleção natural, mutação

genética e reprodução sexuada, dependentes da adaptação dos indivíduos ao meio ambiente[Linden, 2012].

A codificação da solução é uma tradução do problema real para um formato propício para análise computacional. No jargão evolutivo cada solução codificada é chamada de *indivíduo*. Função de avaliação, também chamada de *fitness*, é a forma utilizada por algoritmos evolutivos para avaliar um indivíduo em termos de sua adaptação ou desempenho em relação ao problema abordado. Para um melhor entendimento, pode-se pensar na função como um juiz, que avalia e dá uma nota a um individuo durante a resolução do problema. Este juiz irá avaliar todos os candidatos, dando notas de acordo com o seu desempenho em relação ao problema [Linden, 2012].

O processo evolutivo é promovido pela aplicação de operadores sobre a população de indivíduos. O operador de *Seleção* escolhe dois ou mais indivíduos estocasticamente dentro de uma população para serem recombinados. Existem vários métodos de seleção que são frequentemente usados: roleta, seleção por torneio e seleção elitista [Goldberg, 1989].

Ao escolher dois ou mais indivíduos, aplica-se o operador de *Cruzamento* para gerar novos indivíduos. Desse modo, faz-se uma troca de informação genética entre ambos os pais, gerando assim filhos [Linden, 2012]. O operador de *Mutação* representa a possibilidade de erro durante a cópia de material genético entre pais e filhos, que ocorre com baixa probabilidade, mas é responsável por introduzir características fenotípicas novas na população [Linden, 2012].

3. O problema de Minimização Contínua Irrestrita

Minimização contínua irrestrita é um caso genérico de otimização não-linear, à qual não existe restrições em forma de equações ou em forma de inequações. Porém, restrições de domínio de variáveis podem existir. A questão da restrição de domínio ocorre por conta do conhecimento prévio dos valores que as variáveis podem assumir.

As funções encontradas na literatura [Oliveira e Lorena, 2004] para utilização neste trabalho são as funções Rosenbrock (Ros), Rastrigin (Ras), Schwefel (Sch), Griewank (Gri), Langerman (Lan) e Michalewicz (Mic). Na maioria dessas funções de teste, os limites superiores ${x_i}^{sup}$ e inferiores ${x_i}^{inf}$, que cada variável pode assumir, são definidos anteriormente e são considerados parte do problema. Na Tabela 1 observa-se as funções de teste utilizadas neste trabalho com seus respectivos limites de domínio e soluções ótimas conhecidas.

Tabela 1: Funções de teste utilizadas

Função	**var**	**ótimo**	${x_i}^{inf}$ **;** ${x_i}^{sup}$
Rosenbrock	n	0	-5,12 ; 5,12
Rastrigin	n	0	-5,12 ; 5,12
Schwefel	n	0	-500 ; 500
Griewank	n	0	-600 ; 600
Langerman	5	-1,4	0 ; 10
Langerman	10	-1,4	0 ; 10
Michalewicz	5	-4,687	0 ; π
Michalewicz	10	-9,66	0 ; π

4. *Evolutionary Clustering Search*

A Busca Evolutiva Guiada por Agrupamentos (*Evolutionary Clustering Search* – ECS) propõe a detecção de regiões promissoras baseada na frequência com que são amostrados alguns indivíduos gerados usando o paradigma evolutivo [Oliveira e Lorena, 2004], utilizando um processo de agrupamento de indivíduos para guiar otimizadores locais, como alternativa à aplicação destes operadores apenas sobre a elite da população (abordagem mais popular).

Utiliza-se o termo *cluster* como abstração de um agrupamento de indivíduos referenciados por um indivíduo central. O *ECS* tenta encontrar regiões promissoras através do enquadramento de

subespaços de busca mediante o posicionamento de seus respectivos centros. Pode-se definir um *cluster* pela tripla [Oliveira e Lorena, 2004]:

$$\mathcal{C} = \{\mathbf{c}, r, \mathcal{B}\} \tag{1}$$

onde:

- $\mathbf{c}$ é o centro;
- r é o raio, onde $r \in \mathbb{N}$ e $r > 0$;
- $\mathcal{B}$ é uma estratégia de busca associada ao *cluster*.

O centro $\mathbf{c}$ é um indivíduo (ou solução candidata), que auxilia o *cluster* a representar um subespaço de busca. O raio por sua vez estabelece a distância máxima, a partir do centro, até a qual um indivíduo pode ser associado ao *cluster*. Considerando que $\mathbf{c}$ seja um vetor multidimensional, a região de busca pode ser representada como uma hiperesfera [Oliveira e Lorena, 2004].

No primeiro momento, o centro $\mathbf{c}$ é obtido diretamente do primeiro indivíduo pertencente ao *cluster*. Posteriormente, o centro tende a deslocar progressivamente nas proximidades em direção a regiões onde ocorrem um aumento de atividade. O volume total do *cluster* é definido através do raio r e pode ser calculado de acordo com as características do problema. Um importante requisito é que r deve definir um subespaço de busca apropriado para a estratégia de busca $\mathcal{B}$ [Oliveira e Lorena, 2004].

A métrica de distância está relacionada à estratégia $\mathcal{B}$ e ao espaço de busca. A distância euclidiana, por exemplo, é uma métrica definida no espaço de fenótipos, característicos de problemas de otimização contínua, e que pode também refletir o esforço necessário para se alcançar uma solução localmente ótima a partir do centro de *cluster* [Oliveira e Lorena, 2004].

A estratégia $\mathcal{B}$ é uma sistemática de intensificação de busca, na qual indivíduos de um *cluster* interagem entre si, ao longo do processo de agrupamento, gerando novos indivíduos na mesma região. Podem ser empregadas, ainda como parte de $\mathcal{B}$, heurísticas de melhoria, considerando $\mathbf{c}$ e r como parâmetros, caso a região associada a um dado *cluster* seja considerada promissora [Oliveira e Lorena, 2004]. Em problemas de otimização contínua, r deve ser suficiente para cobrir todo o espaço de busca com um certo número de *clusters*.

4.1. Componentes

O ECS consiste de quatro componentes com atribuições diferentes e conceitualmente independentes. São eles, como mostra a Figura 1: algoritmo evolutivo (AE); agrupador iterativo (AI); analisador de agrupamentos (AA); algoritmo de otimização local (AO).

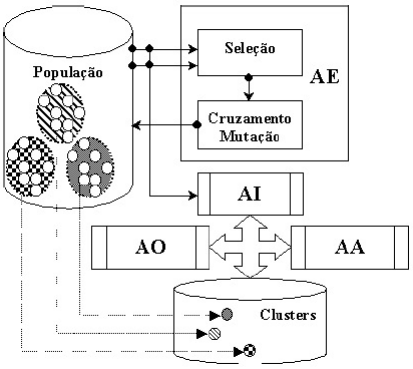


Figura 1: Diagrama conceitual da Busca guiada por Agrupamentos.
Fonte: [Oliveira e Lorena, 2004]

O componente AE é responsável por evoluir a população de indivíduos P, independentemente dos demais componentes. O componente AI copia soluções candidatas de P e realiza o processo de agrupamento, mantendo um centro $\mathbf{c}$ representativo de cada grupo. Tanto indivíduos quanto centros de *cluster* são soluções candidatas em diferentes contextos.

Para evitar esforço extra computacional, o AI realiza o agrupamento de forma incremental, a medida que são geradas em cada iteração regular da AE. Uma métrica de distância deve ser definida, a priori, como uma medida de similaridade para o processo de agrupamento. Além disso, o número de *clusters* deve ser estabelecido para impedir a criação de *clusters* de forma ilimitada [Oliveira e Lorena, 2004].

O componente AA analisa cada *cluster*, em intervalos regulares de gerações, indicando os mais promissores através de uma medida de densidade δ_i, que indica o número de seleções ocorridas recentemente no *cluster* i. Caso algum *cluster* consiga alcançar o limiar δ_i, o módulo AO deve realizar uma busca local a partir do centro c_i. O AA é também responsável pela eliminação de *clusters* com baixa densidade, permitindo assim, a criação de novos centros.

O componente AO implementa um otimizador local que realiza uma intensificação de busca nas supostas regiões promissoras, enquadradas por *clusters* [Oliveira e Lorena, 2004]. Otimizadores locais podem incorporar heurísticas específicas ou algoritmos de subida da encosta (*hill climbing)*. Em geral, são algoritmos que possuem complexidade superior a $O(n)$, sendo n, o tamanho da instância do problema.

4.2. Processo de Agrupamento

Indivíduos, s_k, selecionados para cruzamento são copiados para o componente AI que tenta associá-lo a um dos *clusters* ativos, $\mathbf{c_i}$. Se a distância $\mathfrak{P}$ entre o indivíduo s_k e o centro $\mathbf{c_i}$ for maior que o raio r, um novo *cluster* $\mathbf{c_{new}}$ é criado assumindo s_k como centro. Caso contrário, s_k causa uma perturbação no *cluster*, $\mathbf{c_k}$, mais próximo através de um processo chamado de *assimilação* [Oliveira e Lorena, 2004]. Resumindo-se [Oliveira e Lorena, 2007]:

- $\mathbf{c}_{new} = \mathbf{s}_k$, se $\mathfrak{P}(\mathbf{s}_k, \mathbf{c}_i(t)) > r_i, \forall \mathbf{c_i}(t) \in C$, onde $\mathfrak{P}$ é uma função de similaridade e t é a iteração corrente; ou então
- $\mathbf{c}_i(t+1)$ é resultado da assimilação de $\mathbf{s}_k$ por $\mathbf{c}_i(t)$, de outra forma.

A métrica de distância ou função de similaridade, $\mathfrak{P}$, deve ser consistente com o espaço de busca codificado e regras de vizinhança [Jain e Dubes, 1988], considerando o custo computacional para a obtenção da medida.

Cada vez em que ocorre um processo de *assimilação* em um agrupamento, faz-se com que a *densidade* deste agrupamento aumente. Baseado neste fato, pode-se dizer que existe uma tendência de que agrupamentos em que não ocorrem o mesmo processo se tornem agrupamentos de *baixa densidade* e, por este motivo, tais agrupamentos são eliminados pelo AA.

Ao considerar que na Figura 2 cada ponto dentro de cada agrupamento representa o número de vezes em que este agrupamento foi ativado, pode-se dizer que o agrupamento c_1 é mais denso que o agrupamento c_2 e também que o agrupamento c_1 possivelmente sofrerá uma busca intensificada pelo AO, enquanto o AA poderá eliminar o agrupamento c_2, por este ser considerado de baixa densidade. No fluxograma da Figura 3, pode-se entender melhor o funcionamento do processo de agrupamento.

4.3. Processo de Assimilação

O processo de assimilação é realizado no centro $\mathbf{c}_i(t)$, tendo como referência o indivíduo $\mathbf{s}_k$ selecionado pelo AE. Formalmente pode-se definir a assimilação como:

$$\mathbf{c}_i(t+1) = \mathbf{c}_i(t) \oplus \beta(\mathbf{s}_k \ominus \mathbf{c}_i(t)) \tag{2}$$

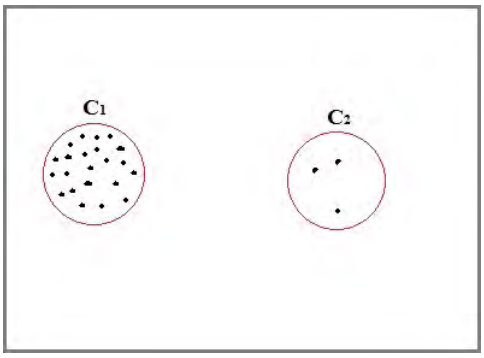


Figura 2: Representação de agrupamentos de alta e baixa densidade

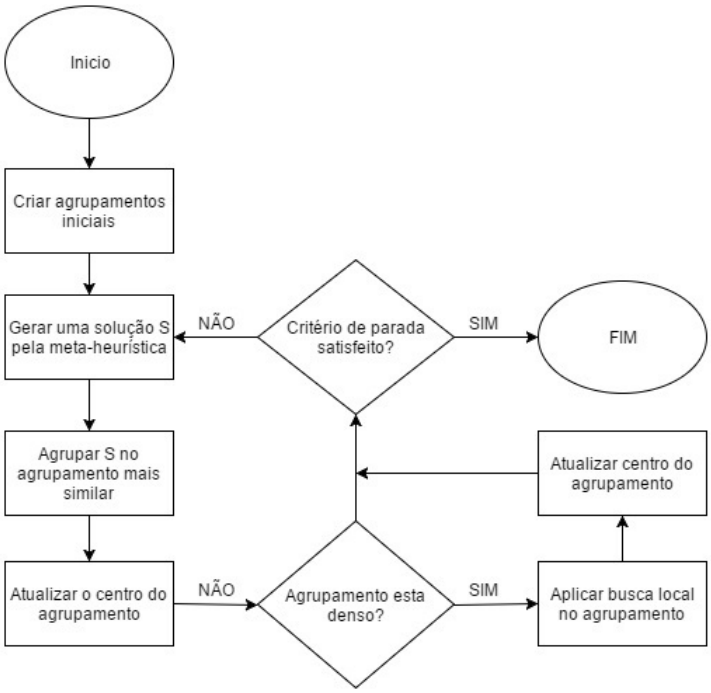


Figura 3: Fluxograma do método de agrupamento.
Fonte: Adaptado de [Chaves, 2009]

onde $\oplus$ e $\ominus$ são operações abstratas. O intervalo $(\mathbf{s}_k \ominus \mathbf{c}_i(t)$ é o vetor de diferenças entre cada uma das n variáveis que compõem as soluções $\mathbf{s}_k$ e $\mathbf{c}_i(t)$, calculado de acordo com alguma métrica de distância [Oliveira e Lorena, 2007]. O valor de β representa um percentual que refere-se ao passo de atualização de $\mathbf{c}_i(t)$ para $\mathbf{c}_i(t+1)$.

Dependendo de β, a assimilação pode adquirir diferentes aspectos como ilustrado na Figura 4.

Na assimilação simples, $\beta \in [0, 1]$ é um parâmetro constante. Isso implica que existe um movimento determinístico de $\mathbf{c}_i(t)$ para $\mathbf{s}_k(t)$. Nota-se, na Figura 4, que apenas um único ponto interno (solução candidata) é gerado próximo à $\mathbf{c}_i(t)$. Quanto maior o β menos conservativo é o movimento. Esse tipo de assimilação pode ser empregado somente com variáveis reais, onde percentual de intervalos podem ser aplicados [Oliveira e Lorena, 2007]. Sua definição formal é:

$$\mathbf{c}_i(t+1) = \mathbf{c}_i(t) + \beta(\mathbf{s}_k - \mathbf{c}_i(t)) \tag{3}$$

Na assimilação por recombinação, um vetor aleatório n-dimensional é usado como parâmetro

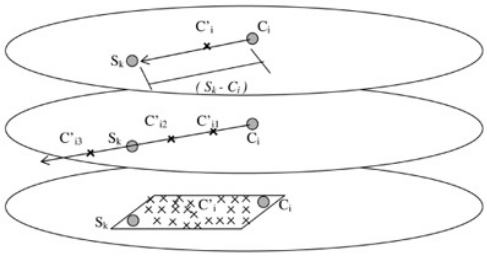


Figura 4: Assimilações simples, por caminho e por recombinação, respectivamente
Fonte: [Oliveira et al., 2013]

de cruzamento entre $c_i(t)$ e s_k. Dessa forma, $c_i(t+1)$ pode assumir uma posição aleatória dentro do hiperplano contendo s_k e $c_i(t)$. Na assimilação por caminho, são gerados vários pontos internos (ou externos em alguns casos), onde cada ponto é uma solução candidata e a melhor avaliada torna-se o novo centro. Nesse tipo de assimilação, β é um vetor n-dimensional de constantes espaçadas uniformemente que conecta o caminho entre $\mathbf{c}_i(t)$ e $\mathbf{s}_k$. Nota-se que cada caminho é avaliado pela função objetivo, gerando um novo centro [Oliveira e Lorena, 2007].

5. Novos assimiladores para otimização contínua

Hipoteticamente, *clusters* têm o papel de modelar um subespaço de busca onde existe atividade de exploração e, eventualmente, intensificação. A modelagem requer três parâmetros: centro, raio e distribuição de probabilidade. Na literatura referente ao *Clustering Search*, admitia-se somente a distribuição uniforme de probabilidade, ou seja, qualquer ponto, a partir do centro, respeitando-se o limite do raio, é equiprovável de ser gerado quando aplicada a estratégia de busca associada ao *cluster*.

A proposta de novos assimiladores modifica a distribuição de probabilidade ao longo do subespaço modelado pelo *cluster*, incluindo novos parâmetros que irão influenciar a estratégia de busca sempre que for agrupada uma nova solução candidata.

Os métodos de assimilação propostos, centro de massa e centro de *cluster*, utilizam novos parâmetros K_1 e K_2 (esses parâmetros são independentes entre si) que devem ser supridos pelo Agrupador Iterativo (AI): os $\{c_k(t-K), c_k(t-K+1), c_k(t-K+2), \cdots, c_k(t), s_k(t)\}$, que são os últimos $K-1$ centros, além da solução s_k em processo de assimilação no instante t.

5.1. Centro de massa

O método do centro de massa é basicamente uma distribuição normal calculada considerando K_1 soluções candidatas, sendo que K_1-1 dessas soluções são os últimos K_1-1 centros de um agrupamento $\{c(t-(K_1-1)), c(t-(K_1-2)), ..., c(t-1), c(t)\}$, incluindo o centro atual, e a última solução da lista é o atual solução s_k sendo assimilada em um dado agrupamento. Nessa situação é calculada a distribuição normal para as K_1 soluções, além de seus respectivos intervalos de confiança através de seu limite inferior L_{inf} e superior L_{sup} com nível de confiança de 95% através dos seguintes cálculos, considerando a média m e a variância v das soluções:

$$m = \left(\left(\sum_{j=t-(K_1-1)}^{t} c(j) \right) + s_k \right) / K_1, \tag{4}$$

$$v = \sum_{j=t-(K_1-1)}^{t} ((c(j)-m)^2/(K_1-1)), \tag{5}$$

e então calcula-se os limites inferiores e superiores do intervalo de confiança:

$$L_{inf} = m - z \times \sqrt{v}/\sqrt{K_1}, \tag{6}$$

$$L_{sup} = m + z \times \sqrt{v}/\sqrt{K_1}, \tag{7}$$

onde z é uma constante que representa o nível de confiança, sendo seu valor igual a $1,96$ que equivale a 95% do intervalo de confiança. Após a geração dos intervalos de confiança, é gerado um novo valor de acordo com a probabilidade dele ser escolhido baseado na curva da distribuição normal, conforme a seguir:

$$h = \sqrt{-2 \ln r_1} \cos(2\pi r_2), \tag{8}$$

em que r_1 e r_2 são vetores independentes cujas coordenadas são geradas através de uma distribuição uniforme dentro do intervalo $[0, 1]$ e após isso tem-se então a geração do novo centro do agrupamento através de:

$$c(t+1) = L_{inf} + (h \times (L_{sup} - L_{inf})). \tag{9}$$

Pode-se observar na Figura 5 o porquê do nome deste método ser Centro de Massa, pois existe uma tendência probabilística de que o novo centro gerado seja em torno do centro de massa das K_1 soluções utilizadas.

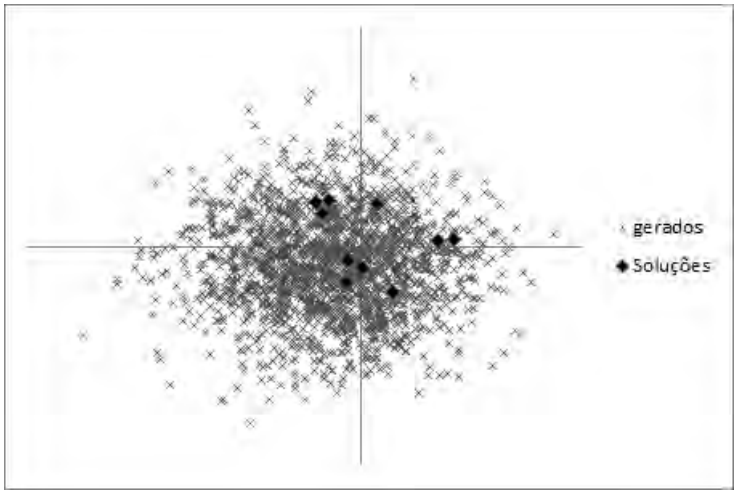


Figura 5: Exemplo de geração de novas soluções através do método Centro de Massa

5.2. Centro de *cluster*

O método do centro de *cluster* é uma adaptação do *parent-centric recombination*(PCX) [Deb et al., 2002], definido por:

$$\vec{y} = \vec{x} + w_\zeta |\vec{d}^{(p)}| + \sum_{\substack{i=1 \\ i \neq p}}^{\mu} w_\eta \bar{D} \vec{e}^{(i)}, \tag{10}$$

no qual μ é o número de soluções candidatas, $\vec{x}$ é uma dessas μ soluções escolhida aleatoriamente entre eles e com igual probabilidade. Calcula-se o $\vec{g}$ como vetor média das μ soluções e estes são utilizados para calcular o vetor direção $\vec{d}^{(p)} = \vec{x}^{(p)} - \vec{g}$.

A seguir, para cada uma das outras $\mu - 1$ soluções, calcula-se a distância perpendicular para a linha $\vec{d}^{(p)}$ e a média dessa distância $\bar{D}$. O vetor $\vec{e}^{(i)}$ são as $\mu - 1$ bases ortonormais que abrangem o subespaço perpendicular a $\vec{d}^{(p)}$. Os parâmetros w_ζ e w_η são normalmente distribuídas com média zero e com variância σ_ζ^2 e σ_η^2, respectivamente, onde têm-se $\sigma_\zeta = 1/\sqrt{\mu - 2}$ e $\sigma_\eta = 0.35/\sqrt{n - \mu - 2}$.

A adaptação feita no PCX para fim de utilizar este método como assimilação foi a alteração de $\vec{x}$ que é um indivíduo aleatório entre os μ (que passa a ser K_2) indivíduos para $c(t)$ que é o centro

atual do agrupamento e passa-se a considerar que $\vec{y}$ será o novo centro no tempo $t+1$, portanto $\vec{y}$ torna-se $c(t+1)$. O $\vec{d}^{(p)}$ passa a ser calculado por $\vec{d}^{(t)} = c(t) - \vec{g}$. Dessa forma tem-se:

$$c(t+1) = c(t) + w_\zeta |\vec{d}^{(t)}| + \sum_{\substack{i=1 \\ i \neq t}}^{K_2} w_\eta \bar{D}\vec{e}^{(i)}. \tag{11}$$

Essa alteração foi feita para que o próximo centro a ser gerado esteja nas proximidades do centro atual, havendo assim movimentações pequenas entre as posições do centro. Esse processo torna a assimilação mais conservadora. Conforme demonstra a Figura 6, é similar à assimilação simples em termos de conservação, contudo observando um campo gaussiano em torno do centro.

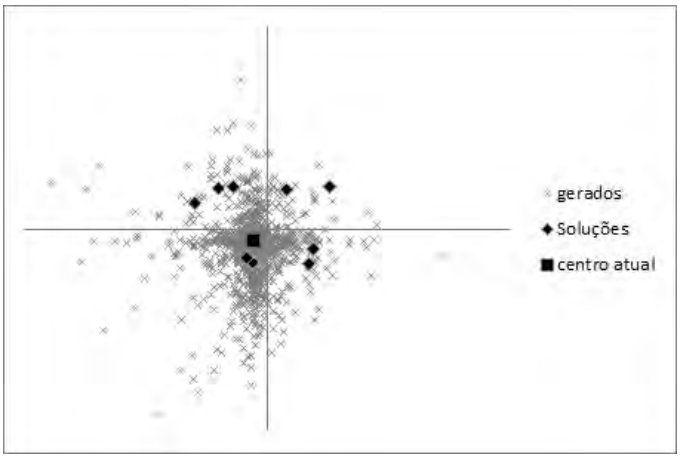


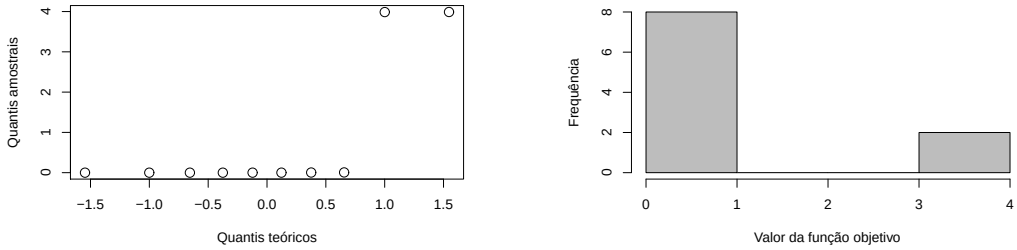
Figura 6: Exemplo de geração de novas soluções através do método centro de *cluster*

6. Resultados

Os testes foram realizados sobre as funções citadas na seção 3, para assimilação simples, por recombinação, centro de massa e centro de *cluster*. Cada execução foi limitada a 10.000 chamadas a função de avaliação do problema. Foram realizados 20 execuções para cada tipo de assimilação, utilizando a mesma população inicial para todos os tipos, porém sendo uma população inicial diferente para cada função de avaliação.

Os valores da função objetivo da última época para as dez execuções foram usados como medida de desempenho. Usou-se o teste estatístico não paramétrico de *Kruskal-Wallis* para encontrar o melhor método, considerando-se que os dados, na maioria dos casos, não foram normalmente distribuídos. A Figura 7 mostra um desses casos, onde visualiza-se primeiramente pelo gráfico *quantil-quantil* e com o teste de *Shapiro-Wilk* ($p < 4.79e-06$) que a distribuição não é normal. Adicionalmente, a distribuição é tão assimétrica que um simples histograma já demonstra esse comportamento. Uma vez detectado uma diferença estatisticamente significante ($p \leq 0,05$) aplicou-se o método de *Dunn* como teste *post-hoc* - com ajuste do valor de p para múltiplas comparações pelo procedimento de *Bonferroni* - uma vez que alguns dados apresentavam *ties*. A análise estatística foi realizada usando a linguagem de programação R, sendo que para implementação do método de *Dunn* usou-se o *Pairwise Multiple Comparison of Mean Ranks Package* (PMCMR) Montgomery [2006].

Os métodos propostos, nomeadamente *centro de cluster* e *massa*, obtiveram um desempenho melhor ou equivalente aos outros métodos usados para comparação, especificamente simples e recombinação. Entretanto, apenas os resultados que foram significativamente melhores, em termos estatísticos, são mostrados neste trabalho. A Figura 8 mostra os resultados, onde o desempenho superior de alguns métodos para a função *Schwefel* - $\chi^2(3) = 18.429, p < 0.000359$ - nomeadamente

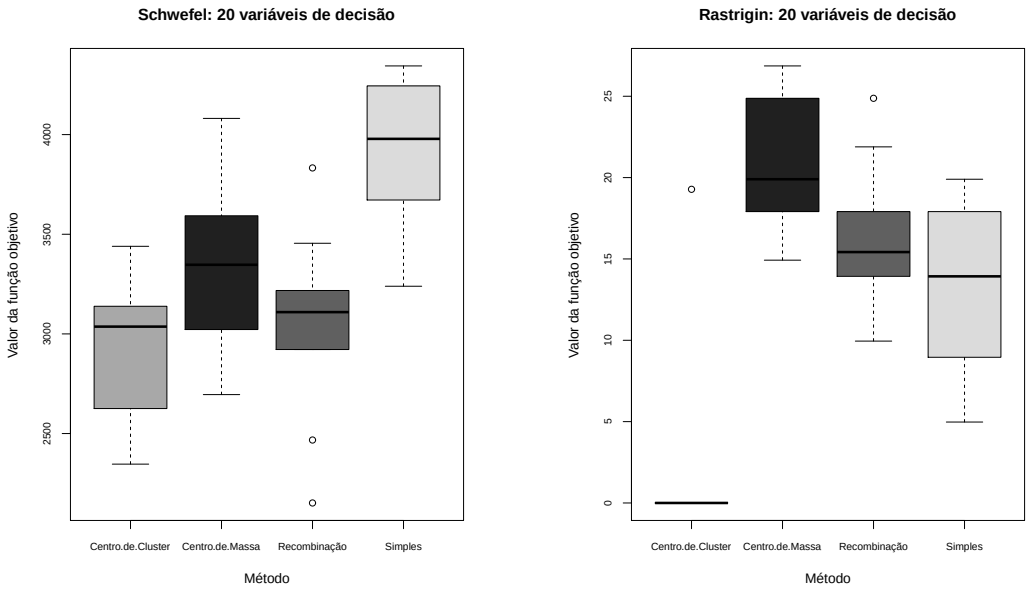


(a) Gráfico quantil-quantil

(b) Histograma

Figura 7: Distribuição dos valores da função objetivo com o método *centro de cluster* para a função *Rosenbrock* com 20 variáveis de decisão. A distribuição não é normal como pode ser visualizado com o gráfico *quantil-quantil*, histograma e usando o teste de *Shapiro-Wilk* ($p < 4.79e - 06$).

centro de cluster e recombinação, enquanto que para a função *Rastrigin* detectou-se uma diferença entre os métodos - $\chi^2(3) = 22.536, p < 5.045e - 05$ - sendo o *centro de cluster* melhor do que alguns dos outros métodos. Adicionalmente, detectou-se uma diferença entre os métodos para a função *Langerman* - $\chi^2(3) = 7.5161, p < 0,05716$ - com valor de p um pouco maior que $0,05$ e com a *post-hoc* análise pode-se afirmar, com um p um pouco maior que 0,05 ($p < 0,057$), que o método centro de cluster foi melhor do que o centro de massa para a função Langerman (Figura 9).



(a) Schwefel - Os métodos centro de cluster e recombinação foram melhores do que o simples ($p < 0.00043$ e 0.00427, respectivamente).

(b) Rastrigin - O método centro de cluster obteve melhor resultado que o recombinação ($p < 0.022$) e centro de massa ($p < 1.8e - 05$).

Figura 8: Funções que foram estatisticamente significante diferentes, com $p < 0.000359$ para a Schwefel e $p < 5.045e - 05$ para a Rastrigin.

7. Conclusão

Hipoteticamente, *clusters* têm o papel de modelar um subespaço de busca onde existe atividade de exploração e, eventualmente, intensificação. Na literatura referente ao *Clustering Search*, admite-se somente a distribuição uniforme de probabilidade, ou seja, qualquer ponto, a partir do

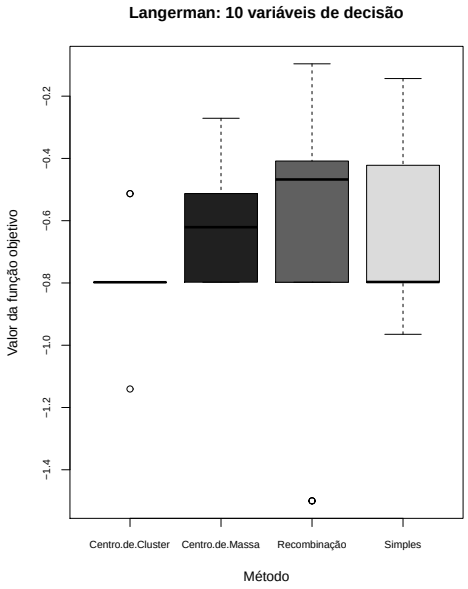


Figura 9: Diferença, com um p um pouco maior que $0,05$ ($p < 0,057$), entre o método centro de cluster e centro de massa para a função Langerman, onde o primeiro obteve um melhor resultado que o segundo.

centro, respeitando-se o limite do raio, é equiprovável de ser gerado quando aplicada a estratégia de busca associada ao *cluster*.

A proposta de novos assimiladores modifica a distribuição de probabilidade ao longo do subespaço modelado pelo *cluster*, incluindo novos parâmetros e melhorando o desempenho como verificado pelo método de centro de *cluster* que foi o método que obteve mais resultados estatisticamente significantes.

Agradecimentos

Os autores agradecem ao CNPq, Proc. 481845/2013-5, e a FAPEMA pelo financiamento parcial desta pesquisa.

Referências

Chaves, A. A. (2009). Uma meta-heurÍstica hÍbrida com busca por agrupamentos aplicada a problemas de otimizacao combinatoria.

Deb, K. et al. (2002). Real-coded evolutionary algorithms with parent-centric recombination. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 1, p. 61–66. IEEE.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition. ISBN 0201157675.

Hansen, N. et al. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18.

Hooke, R. e Jeeves, T. A. (1961). "direct search" solution of numerical and statistical problems. *J. ACM*, 8(2):212–229. ISSN 0004-5411. URL `http://doi.acm.org/10.1145/321062.321069`.

Jain, A. K. e Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA. ISBN 0-13-022278-X.

Linden, R. (2012). *Algoritmos Genéticos*. Ciência Moderna Ltda, Rio de Janeiro, RJ, BR, 3rd edition. ISBN 9788539901951.

Montgomery, D. C. (2006). *Design and Analysis of Experiments*. John Wiley & Sons. ISBN 0470088109.

Oliveira, A. C. M. e Lorena, L. (2007). Hybrid evolutionary algorithms and clustering search. In Abraham, A., Grosan, C., e Ishibuchi, H., editors, *Hybrid Evolutionary Algorithms*, volume 75 of *Studies in Computational Intelligence*, p. 77–99. Springer Berlin / Heidelberg. ISBN 978-3-540-73296-9.

Oliveira, A. C. A. M. d., Chaves, A. A., e Lorena, L. A. N. (2013). Clustering search. *Pesquisa Operacional*, 33:105 – 121. ISSN 0101-7438. URL `http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-74382013000100007&nrm=iso`.

Oliveira, A. C. M. e Lorena, L. A. N. (2004). *Advances in Artificial Intelligence – SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, September 29-Ocotber 1, 2004. Proceedings*, chapter Detecting Promising Areas by Evolutionary Clustering Search, p. 385–394. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN 978-3-540-28645-5.