

Universidade Federal do Maranhão  
Centro de Ciências Exatas e Tecnológicas  
Curso de Ciência da Computação

**ANDRÉ FELIPE DA SILVA OLIVEIRA**

NOVOS MÉTODOS DE ASSIMILAÇÃO BASEADOS EM  
DISTRIBUIÇÃO NORMAL PARA *CLUSTERING SEARCH*

São Luís  
2016

**ANDRÉ FELIPE DA SILVA OLIVEIRA**

NOVOS MÉTODOS DE ASSIMILAÇÃO BASEADOS EM  
DISTRIBUIÇÃO NORMAL PARA *CLUSTERING SEARCH*

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Profº Dr. Alexandre César Muniz de Oliveira

São Luís

2016

---

André Felipe da Silva Oliveira

Novos métodos de assimilação baseados em distribuição normal para *Clustering Search*/ André Felipe da Silva Oliveira. – São Luís, 2016.  
47f.

Orientador: Dr. Alexandre César Muniz de Oliveira

Monografia (Graduação) – Universidade Federal do Maranhão, Curso de Ciência da Computação, 2016.

1. Assimilação. 2. Otimização contínua. 3. Busca guiada por agrupamento  
CDU 004.7

---

André Felipe da Silva Oliveira

## Novos métodos de assimilação baseados em distribuição normal para *Clustering Search*

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

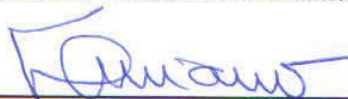
Trabalho aprovado. São Luís, 06 de Abril de 2016:



Prof. Dr. Alexandre César Muniz de  
Oliveira

(Orientador)

Universidade Federal do Maranhão



Prof. Dr. Luciano Reis Countinho

Universidade Federal do Maranhão



Prof. Dr. Paulo Rogério de Almeida  
Ribeiro

Faculdade Laboro

São Luís

2016

*"Dedico este trabalho a todos aqueles que eram fracos e caíram,  
mas se levantaram e tornaram-se fortes."*

# Agradecimentos

Primeiramente gostaria de agradecer aos meus pais, Maria do Socorro Alves da Silva e Evaldo Vieira Oliveira, que sempre me incentivaram e deram suporte para que eu pudesse concluir essa etapa na minha vida. Gostaria de agradecer também as minhas tias, Francisca das Chagas e Isabel Cristina que foram como mães para mim durante todo o período em que estive estudando, tanto no ensino médio quanto no ensino superior. A minha tia Marcia Cristina (e madrinha), Jorge, Antonio Carlos, Claudio, Maria de Fátima e Cleones que sempre me incentivaram e cobraram para que eu pudesse concluir essa fase. Aos meus primos Mayara Cristina, Marcus Antonio e Yury *Mântega* (falecido e amado), que considero como verdadeiros irmãos. A minhas irmãs, Ana Carolina e Rafaela, que sempre estiveram no *meu pé* para poder encerrar essa etapa de uma vez e ao meu irmão, Ramon, que me deu um *choque de realidade* e me ajudou caminhar durante um período complicado da minha vida.

Agradeço a minha amada Thayane Soares, que sempre esteve presente, tanto nos momentos felizes quanto nos momentos mais difíceis, me ajudando a superar inúmeras barreiras.

Agradeço aos meus amigos Hans Newton, Antonio Busson, Ethe Costa, Ruy Oliveira, Marcelo Branco, Alex Newman, Dayson Nascimento e André Luís que me *pertubaram* para finalizar esta monografia e me graduar de vez. Ao meu psiquiatra, João Arnaud, que tem me ajudado a superar a depressão e conseqüentemente ajudando a concluir a monografia.

Agradeço ao meu orientador Alexandre César Muniz de Oliveira<sup>1</sup> que não desistiu de mim e sempre me motivou e ajudou a seguir fazendo a monografia, ao professor *Portela*, que quando coordenador, me deu todo suporte necessário. Agradeço a esta banca de monografia, por terem aceitado o convite e estarem fazendo parte deste momento que é muito especial para mim. Agradeço também aos professores do DEINF<sup>2</sup>(Departamento de Informática) da Universidade Federal do Maranhão<sup>3</sup> que tanto me ensinaram nesses anos de caminhada no curso de Ciência da Computação.

---

<sup>1</sup> Mais informações sobre Alexandre César Muniz de Oliveira em <<http://www.deinf.ufma.br/~acmo/index.html>>

<sup>2</sup> Lista dos professores do DEINF em <[http://www.deinf.ufma.br/index.php?option=com\\_content&view=article&id=46&Itemid=58](http://www.deinf.ufma.br/index.php?option=com_content&view=article&id=46&Itemid=58)>

<sup>3</sup> Mais informações sobre a Universidade Federal do Maranhão <<http://portais.ufma.br/PortalUfma/index.jsf>>

*“Sentia que toda minha vida tinha sido uma espécie de sonho.  
Às vezes eu me perguntava de quem era esse sonho  
e se quem o construiu estava satisfeito.”  
(Douglas Adams, Guia do Mochileiro das Galáxias)*

# 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 então aplicar operadores de busca local nessas regiões.

Os novos métodos de assimilação propostos são chamados de *centro de cluster* e *centro de massa*, ambos baseados em distribuição normal. Os novos métodos foram comparados a propostas anteriores: assimilação simples e por recombinação. O *benchmark* de funções-testes escolhidas para realização dos experimentos computacionais são largamente usadas na literatura. Os resultados encontrados foram satisfatórios e em vários casos apresentaram-se melhores ou equivalentes aos métodos já utilizados na literatura.

**Palavras-chaves:** assimilação. otimização contínua. buscar guiada por agrupamento.



# Abstract

This paper proposes new assimilation methods for Clustering Search for applications related to continuous optimization. For validation, 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 new assimilation methods are called cluster center and center of mass, both based on normal distribution. The new methods were compared to previous proposals: simple and recombination assimilations. The benchmark chosen to perform computational experiments are widely used in the literature. The results were satisfactory and in several cases presented better or equivalent performance when compared to methods already used in the literature.

**Key-words:** assimilation. continuous optimization. Clustering Search.

# Lista de ilustrações

Figura 1 – Vetor representativo de uma codificação binária . . . . .	15
Figura 2 – Vetor representativo de uma codificação com variáveis reais . . . . .	15
Figura 3 – Vetor representativo de uma codificação com variáveis inteiras . . . . .	16
Figura 4 – Indivíduos de uma população e sua respectiva roleta de seleção . . . . .	17
Figura 5 – Representação de uma seleção por torneio . . . . .	18
Figura 6 – Cruzamento . . . . .	18
Figura 7 – Mutação . . . . .	19
Figura 8 – Representação de agrupamentos . . . . .	22
Figura 9 – Diagrama conceitual da Busca guiada por Agrupamentos. . . . .	23
Figura 10 – Representação de agrupamentos de alta e baixa densidade . . . . .	25
Figura 11 – Fluxograma do método de agrupamento. . . . .	26
Figura 12 – Assimilações simples, por caminho e por recombinação, respectivamente	27
Figura 13 – Exemplo de geração de novas soluções através do método Centro de Massa . . . . .	28
Figura 14 – Valores gerados a partir do método centro de cluster . . . . .	30
Figura 15 – Gráfico Boxplot para Ros com 10 variáveis de decisão . . . . .	32
Figura 16 – Gráfico Boxplot para Ros com 20 variáveis de decisão . . . . .	33
Figura 17 – Gráfico Boxplot para Ras com 10 variáveis de decisão . . . . .	35
Figura 18 – Gráfico Boxplot para Ras com 20 variáveis de decisão . . . . .	36
Figura 19 – Gráfico Boxplot para Sch com 10 variáveis de decisão . . . . .	37
Figura 20 – Gráfico Boxplot para Sch com 20 variáveis de decisão . . . . .	38
Figura 21 – Gráfico Boxplot para Gri com 10 variáveis de decisão . . . . .	39
Figura 22 – Gráfico Boxplot para Gri com 20 variáveis de decisão . . . . .	40
Figura 23 – Gráfico Boxplot para Lan com 5 variáveis de decisão . . . . .	41
Figura 24 – Gráfico Boxplot para Lan com 10 variáveis de decisão . . . . .	42
Figura 25 – Gráfico Boxplot para Mic com 5 variáveis de decisão . . . . .	43
Figura 26 – Gráfico Boxplot para Mic com 10 variáveis de decisão . . . . .	44

# Lista de tabelas

Tabela 1 – Funções de teste . . . . .	20
Tabela 2 – Rosenbrock com 10 variáveis de decisão . . . . .	32
Tabela 3 – Rosenbrock com 20 variáveis de decisão . . . . .	33
Tabela 4 – Rastrigin com 10 variáveis de decisão . . . . .	34
Tabela 5 – Rastrigin com 20 variáveis de decisão . . . . .	35
Tabela 6 – Schwefel com 10 variáveis de decisão . . . . .	36
Tabela 7 – Schwefel com 20 variáveis de decisão . . . . .	37
Tabela 8 – Griewank com 10 variáveis de decisão . . . . .	38
Tabela 9 – Griewank com 20 variáveis de decisão . . . . .	39
Tabela 10 – Langerman com 5 variáveis de decisão . . . . .	40
Tabela 11 – Langerman com 10 variáveis de decisão . . . . .	41
Tabela 12 – Michalewicz com 5 variáveis de decisão . . . . .	43
Tabela 13 – Michalewicz com 10 variáveis de decisão . . . . .	44

# Sumário

	<b>Introdução</b>	<b>13</b>
<b>1</b>	<b>ALGORITMOS EVOLUTIVOS</b>	<b>15</b>
1.1	Codificação da solução	15
1.2	Função de avaliação	16
1.3	Mecanismo de seleção	16
1.3.1	Método da roleta	16
1.3.2	Método da seleção por torneio	17
1.3.3	Método da seleção elitista	17
1.4	Mecanismos de cruzamento	18
1.5	Mecanismos de mutação	19
1.6	Algoritmo evolutivo híbrido	19
1.7	O problema de Minimização Contínua Irrestrita	20
<b>2</b>	<b>BUSCA GUIADA POR AGRUPAMENTOS</b>	<b>21</b>
2.1	Formalização do <i>Clustering Search</i>	21
2.1.1	Regiões promissoras e agrupamentos	22
2.1.2	Técnica de agrupamento	22
2.1.3	Componentes	23
2.1.4	Processo de agrupamento	24
2.2	Processo de assimilação	25
2.2.1	Tipos de assimilação	25
2.3	Novos assimiladores para otimização contínua	27
2.3.1	Centro de massa	28
2.3.2	Centro de <i>cluster</i>	29
<b>3</b>	<b>RESULTADOS COMPUTACIONAIS</b>	<b>31</b>
3.1	Resultados obtidos utilizando a função Rosenbrock	31
3.1.1	Resultados para soluções com 10 variáveis de decisão	31
3.1.2	Resultados para soluções com 20 variáveis de decisão	32
3.2	Resultados obtidos utilizando a função Rastrigin	34
3.2.1	Resultados para soluções com 10 variáveis de decisão	34
3.2.2	Resultados para soluções com 20 variáveis de decisão	34
3.3	Resultados obtidos utilizando a função Schwefel	35
3.3.1	Resultados para soluções com 10 variáveis de decisão	36
3.3.2	Resultados para soluções com 20 variáveis de decisão	36

<b>3.4</b>	<b>Resultados obtidos utilizando a função Griewank</b>	<b>37</b>
3.4.1	Resultados para soluções com 10 variáveis de decisão	38
3.4.2	Resultados para soluções com 20 variáveis de decisão	38
<b>3.5</b>	<b>Resultados obtidos utilizando a função Langerman</b>	<b>39</b>
3.5.1	Resultados da função Langerman para soluções com 5 variáveis de decisão	40
3.5.2	Resultados da função Langerman para soluções com 10 variáveis de decisão	41
<b>3.6</b>	<b>Resultados obtidos utilizando a função Michalewicz</b>	<b>42</b>
3.6.1	Resultados da função Michalewicz para soluções com 5 variáveis de decisão	42
3.6.2	Resultados da função Michalewicz para soluções com 10 variáveis de decisão	43
	<b>Conclusão</b>	<b>45</b>
	<b>REFERÊNCIAS</b>	<b>47</b>

# Introdução

Existe uma dificuldade em se encontrar soluções para diversos problemas na sociedade, para isso, se utilizam métodos de Otimização, que tem por objetivo maximizar ou minimizar uma função, tentando assim encontrar o melhor resultado possível para a função dada (para o problema). A otimização pode ser definida basicamente em duas categorias, dependendo de suas variáveis serem contínuas ou discretas. Caso o problema esteja utilizando variáveis discretas, pode-se dizer que ele é um problema de otimização combinatória e caso esteja utilizando variáveis contínuas, tem-se então um problema de otimização contínua que é a categoria de problema que está sendo abordada neste trabalho.

Na literatura existem diversos algoritmos para a resolução de problemas de otimização contínua, sendo o algoritmo *Covariance Matrix Adaptation with Evolution Strategy* (CMA-ES)(HANSEN et al., 2003) o de melhores resultados. Outro algoritmo conhecido e com bons resultados na resolução desse tipo de problema é o *Evolutionary Clustering Search* (ECS)(OLIVEIRA, 2004) que utiliza um algoritmo genético (AG)(GOLDBERG, 1989) que conta também com uma estratégia de busca por agrupamentos tendo como base a ideia de dividir para conquistar.

A busca guiada por agrupamento usa agrupamento de soluções candidatas por características semelhantes. Devido a essa característica, que na prática divide o espaço de busca total em pequenos subespaços onde se concentram soluções candidatas de qualidade, pode-se utilizar algoritmos de busca local para fazer varreduras dentro de cada agrupamento, com ganhos de desempenho e redução de custo computacional. É importante que os centros de agrupamentos estejam sempre atualizados, representando as regiões mais promissoras encontradas pelo algoritmo genético.

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. Otimização contínua é importante pois através dela é possível modelar e resolver problemas que envolvam variáveis reais, tais como problemas que façam parte das áreas da Física, Química, Engenharia, Estatística e Economia.

Neste trabalho, propõe-se utilizar uma técnica de distribuição gaussiana como estratégia de atualização dos centros dos agrupamentos. Até o momento, não existem na literatura, operadores que promovam a atualização de agrupamentos assumindo que a região promissora deve ter um espalhamento gaussiano em torno do centro. Faz-se também uma comparação entre métodos de assimilação já existentes na literatura com as novas assimilações construídas.

A organização deste trabalho é feita de forma em que primeiro (capítulo 1) discute-se sobre algoritmos evolutivos, explicando seus mecanismos. Logo no capítulo 2 discorre-se a respeito da busca guiada por agrupamentos, onde ela é formalizada e têm seus processos e componentes explicados sendo este capítulo finalizado com a explicação dos novos métodos de assimilação que são propostos nesse trabalho. Por último têm-se a análise dos resultados computacionais e a conclusão obtida.

# 1 Algoritmos Evolutivos

Algoritmos evolutivos são algoritmos que se utilizam de modelos computacionais baseados nos processos naturais da evolução para resolver problemas. Existem vários modelos computacionais propostos, mas todos eles têm em comum o conceito de simulação da evolução da espécie, que se utilizam dos processos de seleção, mutação e reprodução, esses processos dependem do desempenho dos indivíduos dentro de um ambiente(LINDEN, 2012).

Neste capítulo trata-se sobre os elementos que compõem os algoritmos evolutivos como no caso dos operadores genéticos, que são aproximações computacionais dos fenômenos evolutivos visto na natureza, exemplos desses fenômenos são a reprodução e a mutação genética.

## 1.1 Codificação da solução

A codificação da solução é uma etapa fundamental no processo de construção das estratégias que serão utilizadas para resolução do problema. Pode-se dizer que a codificação da solução seja uma tradução do problema real para um formato onde a solução possa ser analisada computacionalmente. Um exemplo de codificação é a tradução da solução para um formato de vetor binário, no qual mapeia-se a estrutura do indivíduo do problema em um vetor de  $n$  posições, sendo que  $n$  é o número de variáveis de decisão do problema, cada posição do vetor será tratada como um variável binária, podendo assumir valores 1 ou 0. A Figura 1 demonstra bem a codificação de um vetor binário.

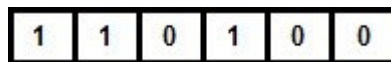


Figura 1: Vetor representativo de uma codificação binária

Assim como a codificação no formato de vetor binário, podem ser utilizados outros tipos de codificações, dependendo do problema e de como ele será tratado. Existem codificações que utilizam variáveis reais e inteiras, é possível observar alguns exemplos de codificação utilizando variáveis reais e inteiras nas Figuras 2 e 3, respectivamente.

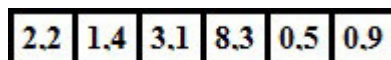


Figura 2: Vetor representativo de uma codificação com variáveis reais





Figura 3: Vetor representativo de uma codificação com variáveis inteiras

## 1.2 Função de avaliação

Função de avaliação é a forma utilizada por algoritmos de otimização para avaliar um indivíduo como solução de um problema abordado. Para um melhor entendimento, pode-se pensar na função como um juiz, que avalia e dá uma nota a um indivíduo na resolução do problema. Este juiz irá avaliar todos os candidatos, dando notas de acordo com o seu desempenho em relação ao problema.

É comum existirem algoritmos otimizadores que sejam genéricos, como os Algoritmos Genéticos. Neste caso é comum que a função de avaliação seja a única ligação verdadeira do programa com o problema real. Em situações como essas, percebe-se que apenas mudando a função de avaliação, também muda-se o problema. Dessa forma, pode-se dizer que a função de avaliação é que determina o problema.

A função de avaliação é a forma numérica pela qual o problema é definido (LINDEN, 2012) dessa forma, a função de avaliação deve ser escolhida com bastante rigor, e nela deve estar embutida todo o conhecimento a respeito do problema a ser resolvido.

## 1.3 Mecanismo de seleção

Mecanismos de seleção são utilizados principalmente por Heurísticas que trabalham com o conceito de população, como no caso dos algoritmos genéticos. Esses mecanismos servem para selecionar indivíduos dentro de uma população, seja de modo determinístico ou de modo não-determinístico.

Existe hoje na literatura um conjunto de métodos de seleção que são frequentemente aplicados, pode-se citar dentre eles os seguintes métodos, aos quais dedicam-se os próximos tópicos: roleta, seleção por torneio e seleção elitista.

### 1.3.1 Método da roleta

Neste método os indivíduos de uma população são tratados como se estivessem em uma roleta que segue uma sequência lógica no qual calcula-se o somatório  $S$  das avaliações dos indivíduos da população, então aleatoriamente se escolhe um número  $k$  entre 0 e  $S$ . Após isso, percorre-se os indivíduos da população um a um, somando suas avaliações de forma acumulativa, quando a soma for maior que  $k$ , então o indivíduo responsável pela última soma que fez com que o resultado da soma fosse maior que  $k$ , é escolhido. Um exemplo de roleta é demonstrado na Figura 4, onde cada fatia do gráfico representa

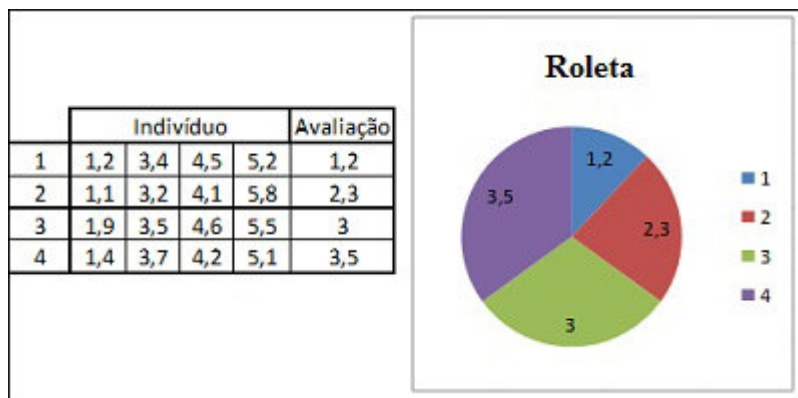


Figura 4: Indivíduos de uma população e sua respectiva roleta de seleção

um indivíduo e o tamanho da fatia representa o tamanho da avaliação do indivíduo em relação ao somatório  $S$ , a contagem da soma acumulativa se realiza a partir do primeiro indivíduo no sentido horário, no momento em que a soma for igual ou superior a  $k$ , então no indivíduo em que a contagem parar é selecionado.

### 1.3.2 Método da seleção por torneio

O método da seleção por torneio se caracteriza por selecionar uma série de indivíduos em uma população e fazer com que eles compitam entre si, utilizando como parâmetro de competição a sua avaliação na função objetivo. Este método tem um parâmetro chamado tamanho do torneio  $k$  que determina o número de indivíduos que podem participar do torneio. É recomendando que não se utilize um  $k$  do tamanho da população, senão o vencedor será sempre o melhor indivíduo da população. Se o  $k$  for muito alto, os  $n - k$  melhores indivíduos predominarão durante o torneio. Na Figura 5 observa-se o modo de funcionamento da seleção por torneio, onde tem-se um torneio com  $k = 2$  e os indivíduos competindo são escolhidos de forma aleatória.

O tamanho mínimo de  $K$  é 2, para que sempre haja no mínimo 2 indivíduos concorrendo. Os indivíduos são escolhidos de forma aleatória, para evitar que o torneio torne-se viciado.

### 1.3.3 Método da seleção elitista

Como percebe-se pelo próprio nome do método, este tipo de seleção sempre escolhe os melhores indivíduos da população, prevenindo a perda desses indivíduos. Quando aplicado a algoritmos evolutivos, como no caso de Algoritmos Genéticos, o desempenho desse método pode ser rapidamente aumentado, pois a seleção elitista garante a permanência dos melhores indivíduos dentro da população.

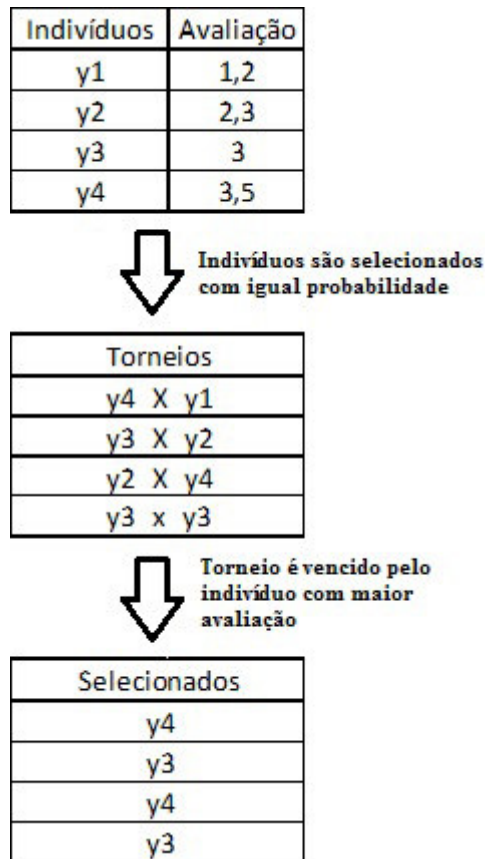


Figura 5: Representação de uma seleção por torneio

### 1.4 Mecanismos de cruzamento

Ao escolher dois indivíduos em uma população é possível recombiná-los para gerar novos indivíduos, desse modo faz-se uma troca de informação genética entre ambos os pais, gerando assim novos indivíduos.

Observa-se, na representação da Figura 6, que dois pais escolhidos tiveram informações trocadas entre eles, fazendo assim com que novos indivíduos fossem gerados.

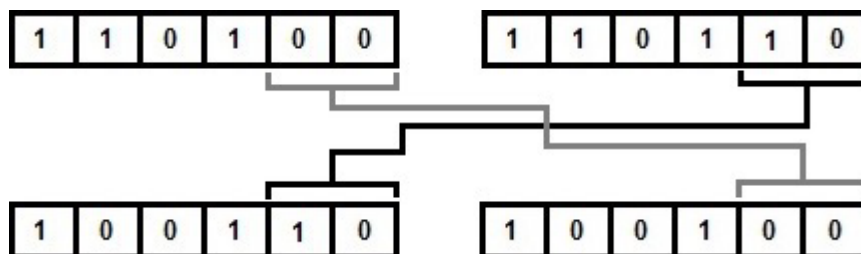


Figura 6: Cruzamento

## 1.5 Mecanismos de mutação

Pode-se aplicar uma chance de que alguma informação (variável de decisão) de um indivíduo da população possa sofrer uma alteração. Essa alteração dar-se-á da forma como for previamente definida, podendo ser de forma aleatória ou probabilística.

Na Figura 7, observa-se que uma informação do indivíduo foi alterada, gerando assim um novo indivíduo.

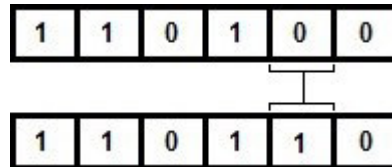


Figura 7: Mutação

## 1.6 Algoritmo evolutivo híbrido

Os algoritmos evolutivos têm uma grande facilidade em percorrer grandes espaços de busca, porém têm dificuldade quando trata-se de intensificar buscas em determinadas regiões (buscas locais), por esse motivo costuma-se associar um algoritmo evolutivo a outras heurísticas com o objetivo de suprir tal necessidade de intensificação em buscas localizadas (OLIVEIRA, 2001).

O maior argumento que favorece o hibridismo está na suposta deficiência que os algoritmos genéticos (que são um tipo de algoritmo evolutivo) têm de intensificação de buscas locais e, dessa forma, os algoritmos genéticos tendem a estabilizarem-se em um ponto onde indivíduos melhores já não são mais encontrados.

Tome-se como exemplo de hibridização o *Evolutionary Clustering Search*, no qual há um Algoritmo Genético associado a uma técnica de busca por agrupamento que utiliza o algoritmo genético como um buscador global dentro de todo espaço e busca, gerando novos indivíduos para população e armazenando esses indivíduos em agrupamentos, utilizando algum critério de semelhança entre esses indivíduos. Quando um agrupamento alcança uma alta frequência de indivíduos por ele assimilados, então é realizada uma busca local intensificada naquele agrupamento, visando encontrar novos indivíduos que sejam bem avaliados. Dessa forma obtém-se um algoritmo genético com alta eficiência de busca em grandes espaços de buscas, associado a uma técnica altamente eficiente em buscas locais. Um algoritmo genético supriu a fraqueza do *clustering search* em buscas globais enquanto o *clustering search* supriu a fraqueza do algoritmo genético em buscas locais.

## 1.7 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.

Tabela 1: Funções de teste

Função	var	ótimo	$x_i^{inf}$ ; $x_i^{sup}$
<b>Rosenbrock</b>	n	0	-5,12 ; 5,12
<b>Rastrigin</b>	n	0	-5,12 ; 5,12
<b>Schwefel</b>	n	0	-500 ; 500
<b>Griewank</b>	n	0	-600 ; 600
<b>Langerman</b>	5	-1,4	0 ; 10
<b>Langerman</b>	10	-1,4	0 ; 10
<b>Michalewicz</b>	5	-4,687	0 ; $\pi$
<b>Michalewicz</b>	10	-9,66	0 ; $\pi$

A questão da restrição de domínio ocorre por conta do conhecimento prévio dos valores que as variáveis podem assumir.

As funções encontradas na literatura para utilização neste trabalho são as funções *Rosenbrock* (*Ros*)(*De Jong, 1975*), *Rastrigin* (*Ras*)(*DIGALAKIS; MARGARITIS, 2002*), *Schwefel* (*Sch*)(*SCHWEFEL, 1981*), *Griewank* (*Gri*)(*DIGALAKIS; MARGARITIS, 2002*), *Langerman* (*Lan*)(*BERSINI et al., 1996*) e *Michalewicz* (*Mic*)(*BERSINI et al., 1996*), estas funções são funções específicas para essa categoria de problemas. 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. Esses limites demarcam o espaço de busca em regiões mais desafiadoras.

Na Tabela 1 observam-se as funções de teste utilizadas neste trabalho com seus respectivos limites de domínio e soluções ótimas conhecidas.

## 2 Busca Guiada por Agrupamentos

Busca Guiada por Agrupamentos, mais conhecida como *Clustering Search*, é construída sobre a ideia basilar de dividir para conquistar, ou seja, tenta-se dividir um problema seguindo um padrão lógico, com vistas a facilitar a resolução do problema.

Ao imaginar-se um conjunto de seres de várias espécies em um grande espaço, encontrar uma determinada espécie dentre tantas em um meio extenso torna-se bastante complicado. Porém pode-se começar a agrupar esses seres por semelhanças, tornando assim mais fácil a tarefa de encontrar a espécie procurada baseando-se nessas características de semelhança que foram utilizadas para agrupar tais seres. Dessa maneira pode-se criar pequenos grupos divididos por características e então concentrar a busca nos locais onde há maior semelhança de características (promissores), evitando assim buscar em regiões que não atendem as características da espécie procurada.

### 2.1 Formalização do *Clustering Search*

*Clustering Search* (CS) é um arcabouço que visa tentar encontrar soluções bem avaliadas e viáveis, usando alguma meta-heurística como geradora da população, para que estes sejam agrupados em termos de codificação (genótipo, tratando-se de Algoritmos Evolutivos), fazendo com que seja gerada uma série de agrupamentos (ou *clusters*). Define-se um agrupamento pela tripla (OLIVEIRA, 2004):

$$C = \langle c, r, \beta \rangle \quad (2.1)$$

Onde  $c$  é o *centro*,  $r$  é o *raio*,  $\beta$  é uma *estratégia de busca* associada ao agrupamento.

O *centro* é uma solução (ou indivíduo) candidata que representa o agrupamento, este centro identifica a localização do agrupamento dentro do espaço de busca, sendo que o *raio* é a distância máxima em relação ao centro em que uma solução pode ser considerada parte deste agrupamento.

Ao observar a Figura 8, pode-se entender melhor o funcionamento de um agrupamento. Observa-se que há  $k$  agrupamentos  $C_k$ , cada um com seus respectivos centros  $c_k$  e raios  $r_k$ . A borda vermelha representa o limite do agrupamento e o que permanece dentro faz parte do agrupamento, este limite é determinado pelo raio e pelo centro.

A *estratégia de busca*  $\beta$  é uma estratégia de busca local onde, através dos indivíduos de um agrupamento e durante o processo de agrupamento, geram novos indivíduos na mesma região. Essa busca local é feita pela interação entre indivíduos dentro de um mesmo agrupamento utilizando-se de movimentos heurísticos, movimento esses que são realizados

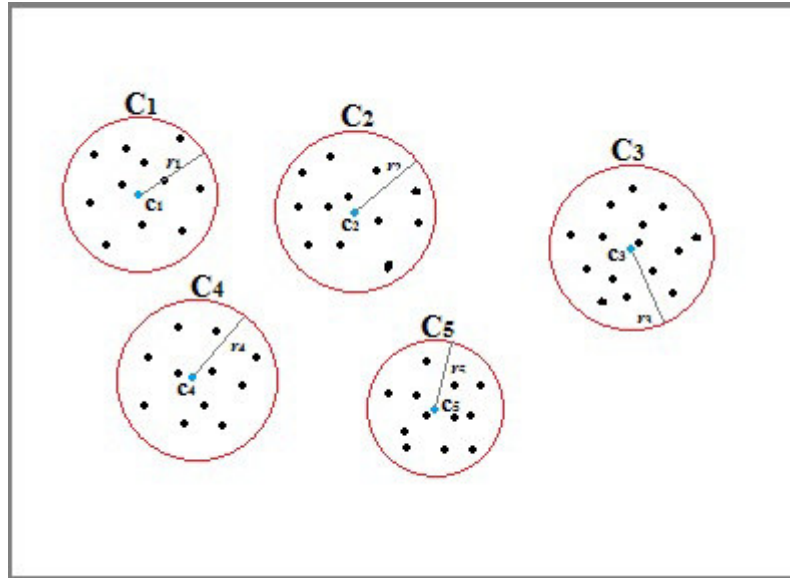


Figura 8: Representação de agrupamentos

quando ocorre a *ativação de agrupamentos* através de funções de seleção ou atualização de um agrupamento. Ainda como parte de  $\beta$ , pode-se aplicar heurísticas de melhorias, tendo como parâmetros  $c$  e  $r$ , desde que a região associada e o agrupamento em questão sejam considerados uma região promissora.

### 2.1.1 Regiões promissoras e agrupamentos

Existe uma tendência de que indivíduos bem avaliados se localizarem em regiões próximas entre si, criando grupos de regiões dentro do espaço de busca que são consideradas regiões promissoras, ou seja, regiões com maior chance de encontrarem-se bons indivíduos ou até mesmo o resultado ótimo. A busca guiada por agrupamentos se utiliza desta tendência para concentrar suas buscas dentro desses grupos promissores, evitando assim esforços computacionais excessivos em locais com baixa chance de encontrarem-se resultados satisfatórios.

Considera-se regiões promissoras um subespaço dentro do espaço de busca, onde foram encontrados indivíduos viáveis e bem avaliados. A verificação para determinar se uma região é promissora se faz através da frequência em que se encontram indivíduos bem avaliados dentro daquela região.

### 2.1.2 Técnica de agrupamento

Ao iniciar um agrupamento, o centro  $c$  é encontrado de forma aleatória, mas ele tende a se movimentar progressivamente para regiões próximas dentro do espaço de busca que sejam realmente promissoras. O tamanho (volume) do agrupamento é dado pelo raio

$r$  e este pode ser calculado de acordo com a natureza do problema. É importante que o subespaço de busca definido por  $r$  seja apropriado para aplicação da estratégia de busca  $\beta$ .

A busca guiada por agrupamento pode manter agrupamentos com volumes diferentes entre si e também utilizar estratégias diferentes em cada agrupamento, dependendo do subespaço de busca em que ele se encontra. Pode-se utilizar uma mesma estratégia de busca em todos os agrupamentos, ou utilizar diferentes estratégias, dependendo dos objetivos desejados.

### 2.1.3 Componentes

Conforme a Figura 9, observa-se que a busca guiada por agrupamentos é composta por componentes no qual:

AI – agrupador interativo;

AA – analisador de agrupamentos;

AO – algoritmo de otimização local.

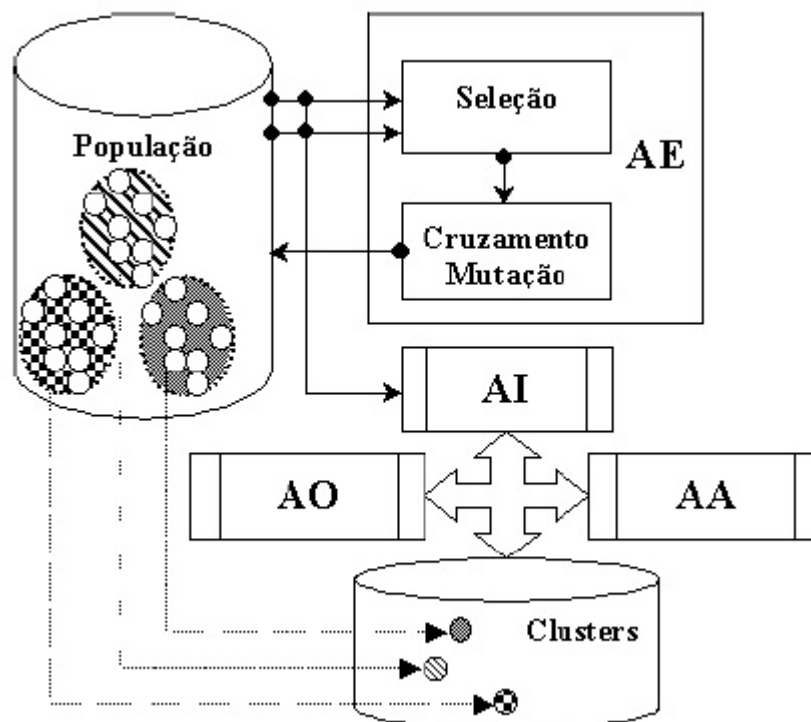


Figura 9: Diagrama conceitual da Busca guiada por Agrupamentos.

Fonte: Acervo do autor (OLIVEIRA, 2004)

O AI tem a função de agrupar interativamente novas soluções candidatas, onde essas soluções são encontradas através de buscas locais executadas a cada vez que um novo indivíduo é incluído no agrupamento. Essas novas soluções candidatas encontradas



são adicionadas ao agrupamento, porém não podem ser chamadas de indivíduos, pois não fazem parte da população gerada pela heurística.

O AA tem a função de verificar cada agrupamento, fazendo isso em intervalos regulares de gerações da heurística, esse processo indica que agrupamentos são considerados promissores. Para determinar quando um agrupamento é promissor, utiliza-se um conceito de densidade, conceito esse que é utilizado para medir o número de vezes que um agrupamento é ativado quando é incluída uma nova solução candidata. O AA também tem a função de eliminar os agrupamentos com baixa densidade.

O AO tem a função de intensificar a busca em supostas regiões promissoras, que são os agrupamentos com alta taxa de ativação (onde vários indivíduos foram agrupados). Este processo ocorre sempre que o AA encontra um novo agrupamento alvo.

#### 2.1.4 Processo de agrupamento

O AI é o núcleo da busca guiada por agrupamento, cada vez que a heurística de busca encontra uma nova solução candidata viável, esta possui uma localização dentro do espaço de busca, agrupa-se essa solução em um agrupamento próximo, caso ele esteja dentro do raio  $r$  de alcance em relação à solução candidata que representa o centro deste agrupamento, neste caso essa nova solução causa uma perturbação ao centro do agrupamento (chama-se essa perturbação de *assimilação*) provocando a atualização do centro com a nova informação recebida. Caso a nova solução candidata encontrada não esteja dentro do raio de alcance de pelo menos um dos agrupamentos (ou seja, uma solução considerada suficientemente nova), essa solução torna-se o centro de um novo agrupamento.

Cada vez em que ocorre um processo de *assimilação* em um agrupamento, faz-se com que a *densidade* deste agrupamento aumente. Baseado-se 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 10 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 11, pode-se entender melhor o funcionamento do processo de agrupamento.

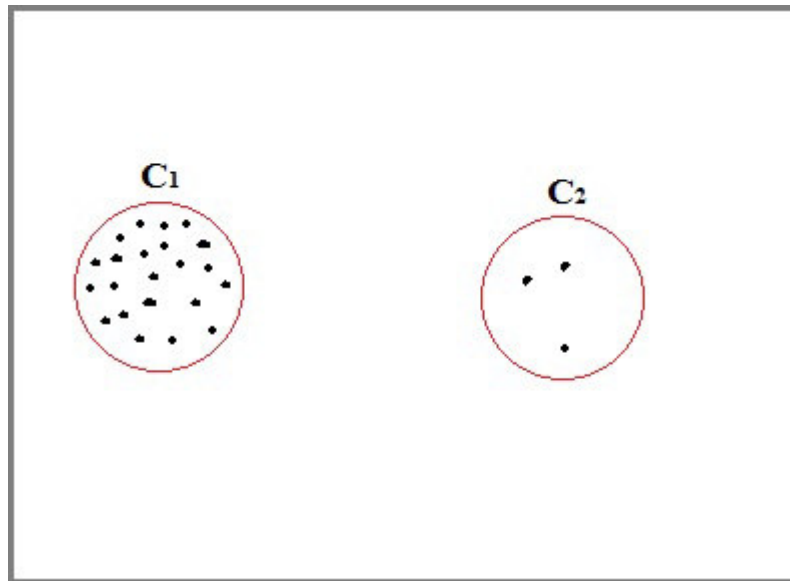


Figura 10: Representação de agrupamentos de alta e baixa densidade

## 2.2 Processo de assimilação

O processo de *assimilação* é aplicado quando uma nova solução candidata é inserida em um agrupamento. Aplica-se um processo no centro do agrupamento,  $c_i$ , considerando-se a solução candidata inserida,  $s_k$ , gera-se uma atualização do centro do agrupamento,  $c'_i$ . A partir desta ideia pode-se dizer que (OLIVEIRA, 2004):

$$c'_i = c_i \oplus \theta(s_k \ominus c_i) \quad (2.2)$$

Onde considera-se  $\oplus$  e  $\ominus$  como operações abstratas aplicadas sobre as soluções candidatas  $c_i$  e  $s_k$ , operações essas que significam, respectivamente, soma e subtração de soluções. O intervalo  $(s_k \ominus c_i)$  seja o vetor de diferenças entre cada uma das  $n$  variáveis de decisão que fazem parte das soluções candidatas  $c_i$  e  $s_k$ , cujo intervalo é calculado de acordo com alguma métrica de distância previamente determinada.

Uma quantidade percentual  $\theta$  do valor do intervalo  $(s_k \ominus c_i)$  é utilizada para atualizar o centro  $c_i$ , gerando dessa forma um novo centro  $c'_i$ . A *assimilação* pode assumir diferentes formas, dependendo de  $\theta$ .

### 2.2.1 Tipos de assimilação

A assimilação pode ser efetuada de diversas formas, neste trabalho discute-se a respeito de três tipos principais, que são as assimilações *simples*, *por recombinação* e *por caminho*.

Pode-se caracterizar a *assimilação simples* quando  $\theta \in [0, 1]$ , fazendo com que o centro  $c_i$  se mova em direção à solução candidata  $s_k$  de forma constante podendo ser

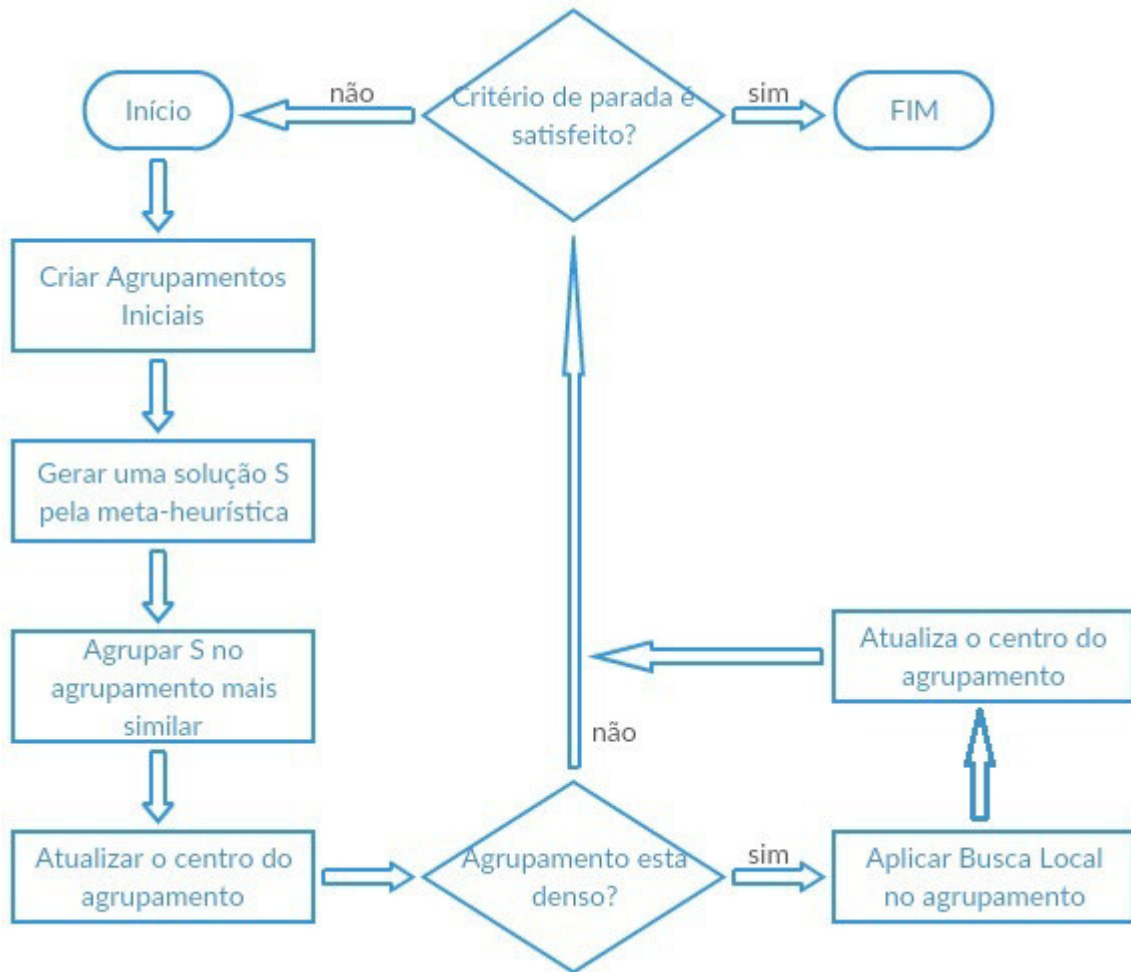


Figura 11: Fluxograma do método de agrupamento.

Fonte: Adaptado de (CHAVES, 2009)

mais ou menos conservador (considera-se conservador quando o novo centro move-se para distâncias próximas ao centro atual, evitando movimentos muito brusco e dessa forma perdendo menos informações do agrupamento a cada atualização). Na *assimilação por caminho* tem-se que  $\theta$  é descrito como um vetor de parâmetros constantes e distintos, no qual  $\theta_k \in [0, 1]$ , sendo que cada elemento é utilizado para gerar uma nova solução candidata entre  $c_i$  e  $s_k$ , que por sua vez, cada solução gerada é avaliada segundo a função de avaliação. Já na *assimilação por recombinação* tem-se um vetor aleatório  $n$ -dimensional que é utilizado como parâmetro de cruzamento entre as soluções  $c_i$  e  $s_k$ , através disso é possível obter-se  $c'_i$  de uma posição aleatória interna ao hiperplano que contem  $c_i$  e  $s_k$ . Na Figura 12, pode-se observar uma demonstração de como esses três métodos funcionam.

Analisando os processos de *assimilação simples* e *assimilação por recombinação*, percebe-se que eles geram um ponto interno ao agrupamento que deve ser avaliado a seguir. Não havendo nenhuma garantia de que esse ponto seja o melhor ponto dentro do agrupamento. Entretanto, na *assimilação por caminho*, obtém-se um conjunto de pontos

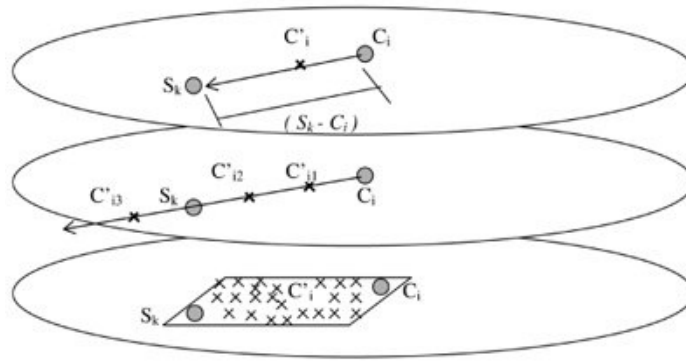


Figura 12: Assimilações simples, por caminho e por recombinação, respectivamente

Fonte: Acervo do autor (OLIVEIRA et al., 2013)

que podem tanto ser internos (interpolação), como podem ser externos (extrapolação) e todos devem ser avaliados. Levando em consideração o número de chamadas que a *assimilação por caminho* fará a função de avaliação, percebe-se o quão custoso ela pode ser, porém existe uma aparente vantagem em manter a melhor solução candidata encontrada, dentro do agrupamento, como centro.

Os movimentos de extrapolação podem conduzir o centro para fora dos limites do agrupamento, dessa forma, torna-se necessária a existência de mecanismos que evitem que agrupamentos se sobreponham e caso sejam sobrepostos, um deles seja eliminado. A eliminação entre agrupamentos sobrepostos pode ser interpretada como uma forma de competição entre eles. Os indivíduos que estão sendo agrupados, devem ser agrupados ao agrupamento mais recente entre eles, fazendo com que os agrupamentos mais recentes adquiram novos indivíduos e fazendo o agrupamento mais antigo perder densidade, sendo eliminado posteriormente.

## 2.3 Novos assimiladores para otimização contínua

Observando a forma como funciona os métodos de assimilação, percebe-se que é possível melhorar a eficiência do *clustering search* melhorando o método de assimilação. Baseando-se nesta ideia foram desenvolvido dois novos métodos de assimilação para serem aplicados no *clustering search* a fim de comparar a eficácia junto aos métodos de assimilação simples e por recombinação. Esses novos métodos são baseados na distribuição normal e utilizam os  $k - 1$  últimos centros de um agrupamento mais a atual solução candidata sendo assimilada neste agrupamento e dessa forma procura-se um novo centro de forma mais precisa e probabilística.

### 2.3.1 Centro de massa

O método do centro de massa é basicamente uma distribuição normal calculada considerando  $k$  soluções candidatas, sendo que  $k - 1$  dessas soluções são os últimos  $k - 1$  centros de um agrupamento, incluindo o centro atual, e a última solução da lista é o atual solução sendo assimilada neste agrupamento, nessa situação é calculada a distribuição normal para cada variável de decisão  $\{x_{n1}, x_{n2}, \dots, x_{nk}\}$  dessas  $k$  soluções e então através da curva de probabilidade gerada para cada variável de decisão, gera-se novos valores para a representante dessas variáveis em uma nova solução candidata e esta solução será o novo centro do agrupamento.

A fórmula da distribuição normal é dada por:

$$f(x) = \frac{e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}}{\sigma\sqrt{2\pi}} \quad (2.3)$$

No qual  $\mu$  representa a média populacional e  $\sigma$  representa o desvio-padrão populacional.

Pode-se observar na Figura 13 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  $k$  soluções utilizadas.

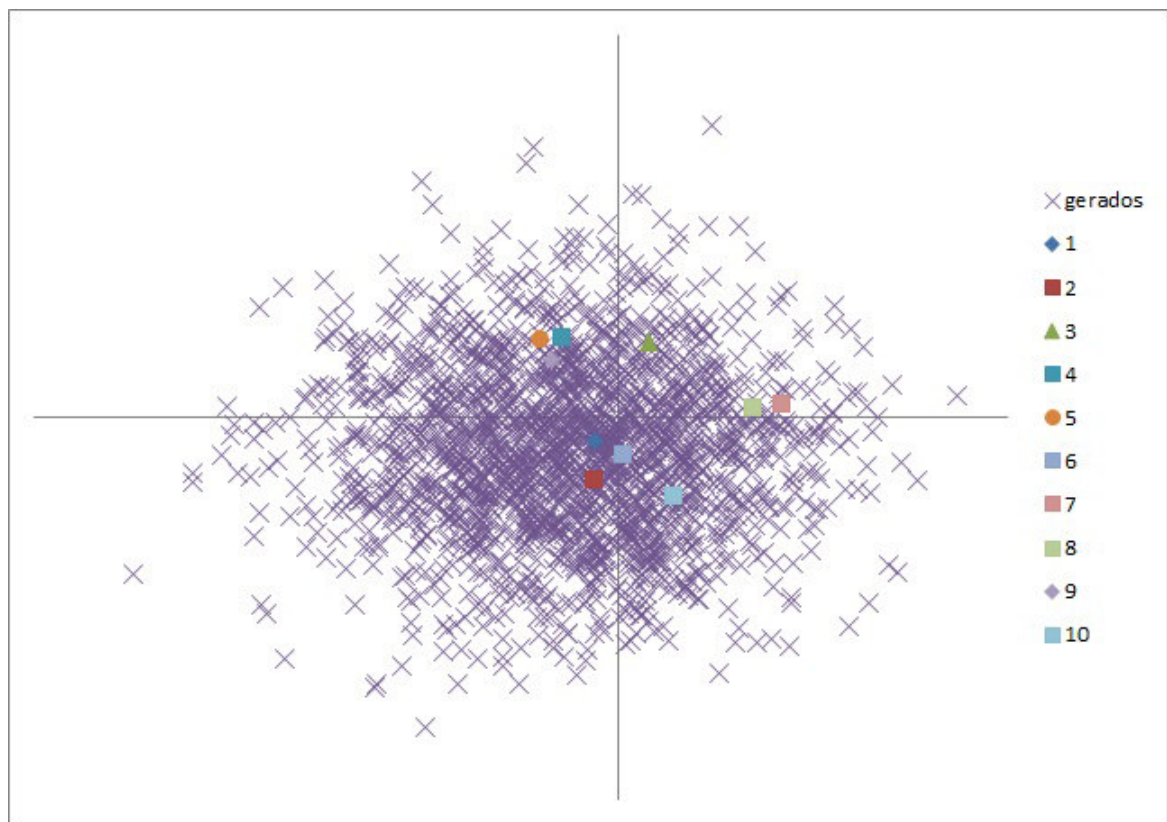


Figura 13: Exemplo de geração de novas soluções através do método Centro de Massa

### 2.3.2 Centro de *cluster*

O método do centro de *cluster* é uma adaptação do *parent-centric recombination*(PCX)(DEB et al., 2002), tem-se a definição do PCX:

$$\vec{y} = \vec{x} + w_\zeta |d^{\vec{p}}| + \sum_{\substack{i=1 \\ i \neq p}}^{\mu} w_\eta \bar{D} \vec{e}^{(i)}, \quad (2.4)$$

No qual  $\mu$  é o número de soluções candidatas,  $\vec{x}$  é uma dessas  $\mu$  soluções escolhida aleatoriamente entre eles e com igual probabilidade. Calcula-se o  $\vec{g}$  como vetor média das  $\mu$  soluções e estes são utilizados para calcular o vetor direção  $d^{\vec{p}} = \vec{x}^{(p)} - \vec{g}$ . A seguir, para cada uma das outras  $\mu - 1$  soluções, calcula-se a distância perpendicular para a linha  $d^{\vec{p}}$  e a média dessa distância  $\bar{D}$ . O  $\vec{e}^{(i)}$  são as  $\mu - 1$  bases ortonormais que abrangem o subespaço perpendicular a  $d^{\vec{p}}$ . Os parâmetros  $w_\zeta$  e  $w_\eta$  são variáveis normalmente distribuídas com média zero e com variância  $\sigma_\zeta^2$  e  $\sigma_\eta^2$ , respectivamente, onde tem-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  $\mu$  indivíduos para  $\vec{c}$  que é o centro atual do agrupamento. Dessa forma tem-se:

$$\vec{y} = \vec{c} + w_\zeta |d^{\vec{p}}| + \sum_{\substack{i=1 \\ i \neq p}}^{\mu} w_\eta \bar{D} \vec{e}^{(i)}. \quad (2.5)$$

Essa alteração foi feita para que o próximo centro a ser gerado, seja em torno do centro atual, havendo assim movimentações pequenas entre as posições do centro. Esse processo torna a assimilação mais conservadora, dificultando mudanças bruscas e muita perda de informação do agrupamento, isso é feito para evitar que sejam perdida muitas informações rapidamente sem que antes essas informações tenham sido exaustivamente exploradas.

A Figura 14 demonstra bem como a assimilação baseada no centro de cluster mantém um bom conservadorismo em relação aos movimentos do centro do agrupamento, a probabilidade da posição do centro atual mudar para uma posição considerada perto é muito maior do que a probabilidade do centro se mover em grandes distâncias.

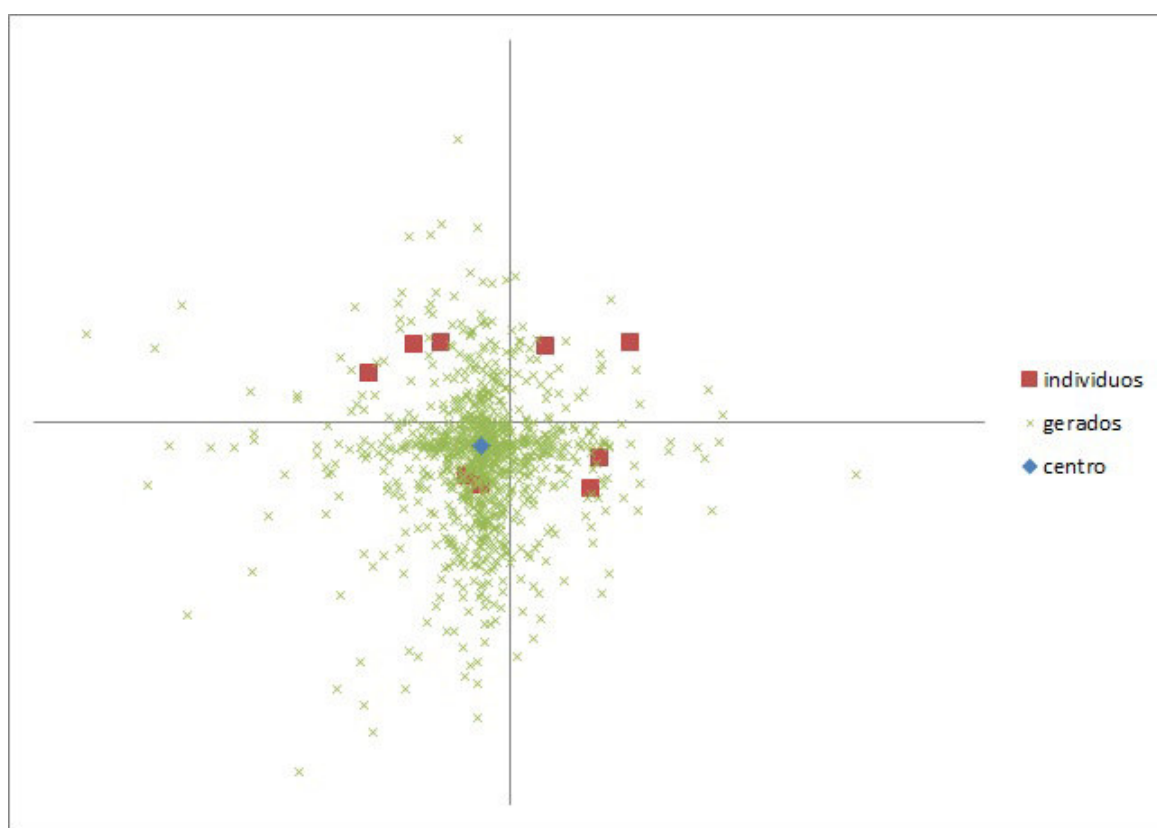


Figura 14: Valores gerados a partir do método centro de cluster

## 3 Resultados Computacionais

Os testes foram realizados usando um *benchmark* constituído das funções: *Rosenbrock* (*Ros*)(*De Jong, 1975*), *Rastrigin* (*Ras*)(*DIGALAKIS; MARGARITIS, 2002*), *Schwefel* (*Sch*)(*SCHWEFEL, 1981*), *Griewank* (*Gri*)(*DIGALAKIS; MARGARITIS, 2002*), *Langerman* (*Lan*)(*BERSINI et al., 1996*) e *Michalewicz* (*Mic*)(*BERSINI et al., 1996*). Mais informações sobre esse benchmark pode ser encontrado em (*OLIVEIRA, 2004*).

Nos experimentos, foram utilizadas assimilações do tipo simples, recombinação, centro de massa e centro de *cluster*. Nos tópicos a seguir, são demonstradas as avaliações e significância obtidas através da comparação desses quatro tipos de assimilações. Foram também realizadas 28 execuções para cada função, para cada execução um limite de até 10000 chamadas a função objetivo e um máximo de até 180000 gerações.

As assimilações utilizando centro de massa, foram feitas utilizando o parâmetro  $k$ , que determina o numero de soluções candidatas utilizadas na distribuição normal, é fixa em 20 para todas as funções, chegou-se a esse valor por ter-se obtido melhor desempenho com ele. Já a assimilação com centro de *cluster* utiliza-se como parâmetro  $k$ , que determina o número de soluções utilizadas no seu cálculo, o valor 10, que também foi o valor que permitiu o melhor desempenho deste método. A escala dos gráficos apresentados nos próximos tópicos é dada pelos resultados da função de avaliação.

### 3.1 Resultados obtidos utilizando a função Rosenbrock

A função *Rosenbrock* é dada por:

$$f_{Ros}(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2) \quad (3.1)$$

$$(-5, 12 \leq x_i < 5, 12)$$

$$\min(f_{Ros}(x)) = f(1, 1, \dots, 1) = 0.$$

Os testes realizados foram feitos considerando soluções com 10 e 20 variáveis de decisão.

#### 3.1.1 Resultados para soluções com 10 variáveis de decisão

Pode-se observar que os resultados obtidos na função *Ros* com 10 variáveis de decisão a partir da Tabela 2, que a assimilação Centro de *cluster* obteve melhores resultados que as demais através da comparação da média dos resultados obtidos nos testes e a



assimilação centro de massa obteve melhores resultados que as assimilações simples e por recombinação.

Para corroborar os resultados obtidos na Tabela 2, pode-se analisar o gráfico *Boxplot* da Figura 15 que compara visualmente as taxas de variação dos resultados. Por o ótimo da função *Ros* ser 0 para quaisquer  $n$  variáveis, de acordo com a Tabela 1, pode-se então afirmar que quem é visualmente superior é a assimilação que está mais próxima ao valor ótimo e menor variação de resultados, a prioridade é para proximidade ao ótimo e depois a variação de resultados e isso é representando pelas caixas no gráfico.

Tabela 2: Rosenbrock com 10 variáveis de decisão

Assimilações	Média de melhores resultados	Média de chamadas à FO
<b>Simple</b> s	0,000617107	3959,714286
<b>Recombinação</b>	0,000685496	2922,321893
<b>Centro de massa</b>	0,000549357	4545,428571
<b>Centro de cluster</b>	0,000309714	4782,785714

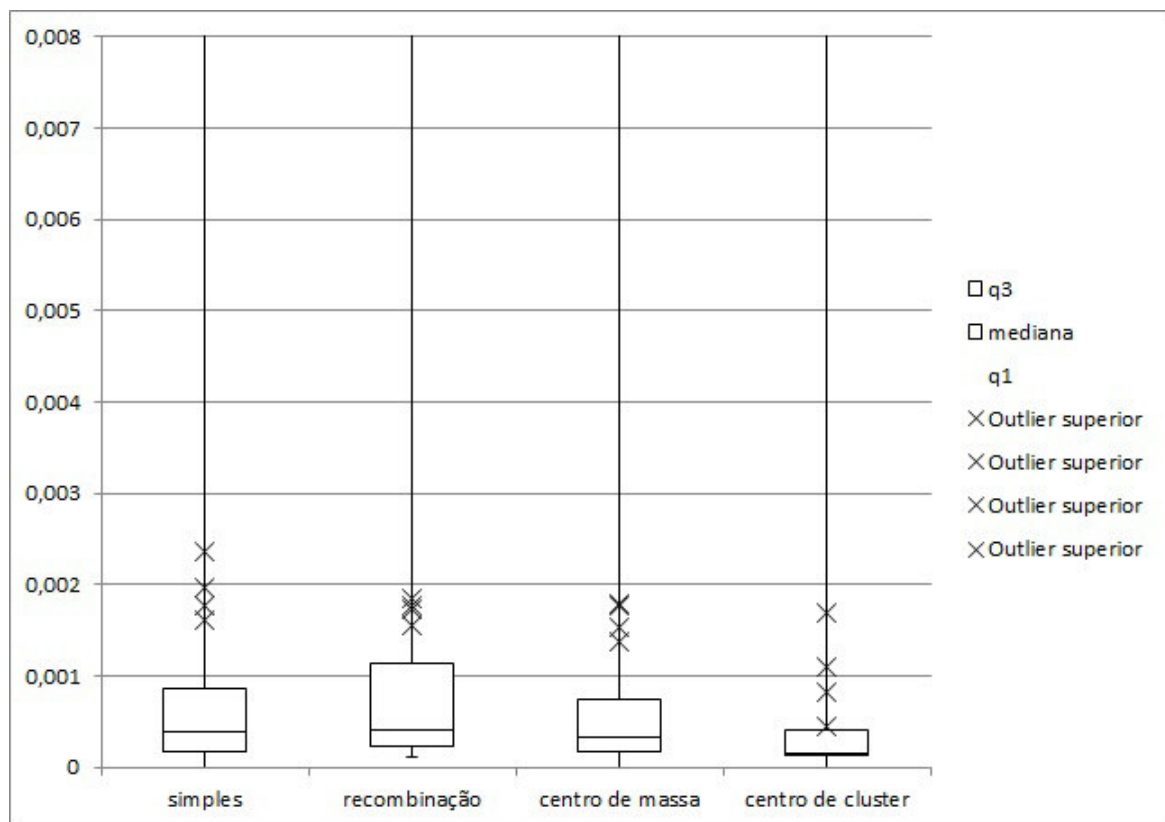


Figura 15: Gráfico Boxplot para Ros com 10 variáveis de decisão

### 3.1.2 Resultados para soluções com 20 variáveis de decisão

Com resultados para *Ros* com 20 variáveis de decisão, observando a Tabela 3, a assimilação centro de *cluster* continuou obtendo melhores resultados no geral, enquanto

o centro de massa teve uma queda em relação as assimilações simples e recombinação. Entretanto, na comparação do gráfico *boxplot* da Figura 16, pode-se dizer que a assimilação por centro de massa tem é visualmente superior do que a por recombinação, isso por conta da variação de resultados da assimilação por centro de massa ser imensamente inferior a da recombinação.

Nesta situação, até mesmo na comparação dos gráficos, a assimilação por centro de *cluster* continuou tendo desempenho melhor que as demais assimilações.

Tabela 3: Rosenbrock com 20 variáveis de decisão

Assimilações	Média de melhores resultados	Média de chamadas à FO
<b>Simple</b> s	0,856380571	7302
<b>Recombinação</b>	1,035833964	8337,464286
<b>Centro de massa</b>	1,377823857	9158,035714
<b>Centro de cluster</b>	0,685293679	9143,071429

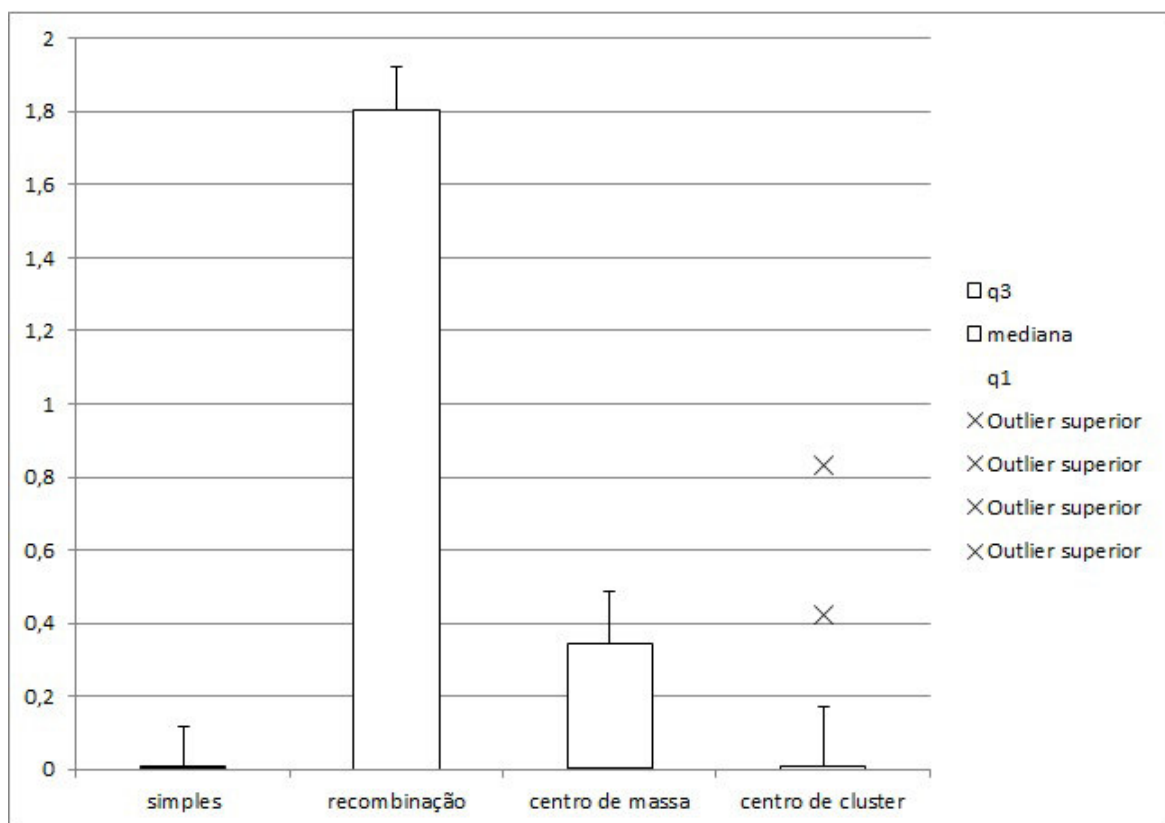


Figura 16: Gráfico Boxplot para Ros com 20 variáveis de decisão

## 3.2 Resultados obtidos utilizando a função Rastrigin

A função *Rastrigin* é dada por:

$$f_{Ras}(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)) \quad (3.2)$$

$$(-5, 12 \leq x_i < 5, 12)$$

$$\min(f_{Ras}(x)) = f(0, 0, \dots, 0) = 0.$$

Os testes realizados foram feitos considerando soluções com 10 e 20 variáveis de decisão.

### 3.2.1 Resultados para soluções com 10 variáveis de decisão

Para a função *Ras* com 10 variáveis, a assimilação centro de *cluster* obteve melhores resultados tanto na comparação das médias (Tabela 4), quanto no gráfico *boxplot* (Figura 17), sendo visualmente superior, inclusive tendo sido o único a encontrar o ótimo que é 0 para quaisquer  $n$  variáveis, de acordo com a tabela 1, em 24 das 28 execuções. O centro de massa obteve um desempenho bem inferior aos outros métodos, tanto na comparação das médias, quanto visualmente no gráfico da Figura 17.

Tabela 4: Rastrigin com 10 variáveis de decisão

Assimilações	Média de melhores resultados	Média de chamadas à FO
<b>Simples</b>	4,116004286	10007,14286
<b>Recombinação</b>	4,068325571	10008,14286
<b>Centro de massa</b>	5,245402893	10007
<b>Centro de cluster</b>	0,035552321	4940,428571

### 3.2.2 Resultados para soluções com 20 variáveis de decisão

Para a função *Ras* com 20 variáveis, a assimilação centro de *cluster* também obteve melhores resultados tanto na comparação das médias (Tabela 5), quanto visualmente no gráfico da Figura 18, inclusive tendo também sido o único a encontrar o ótimo, o ótimo foi encontrado em 14 das 28 execuções. O centro de massa obteve um desempenho bem inferior aos outros métodos, tanto na comparação das médias, quanto na comparação do gráfico.

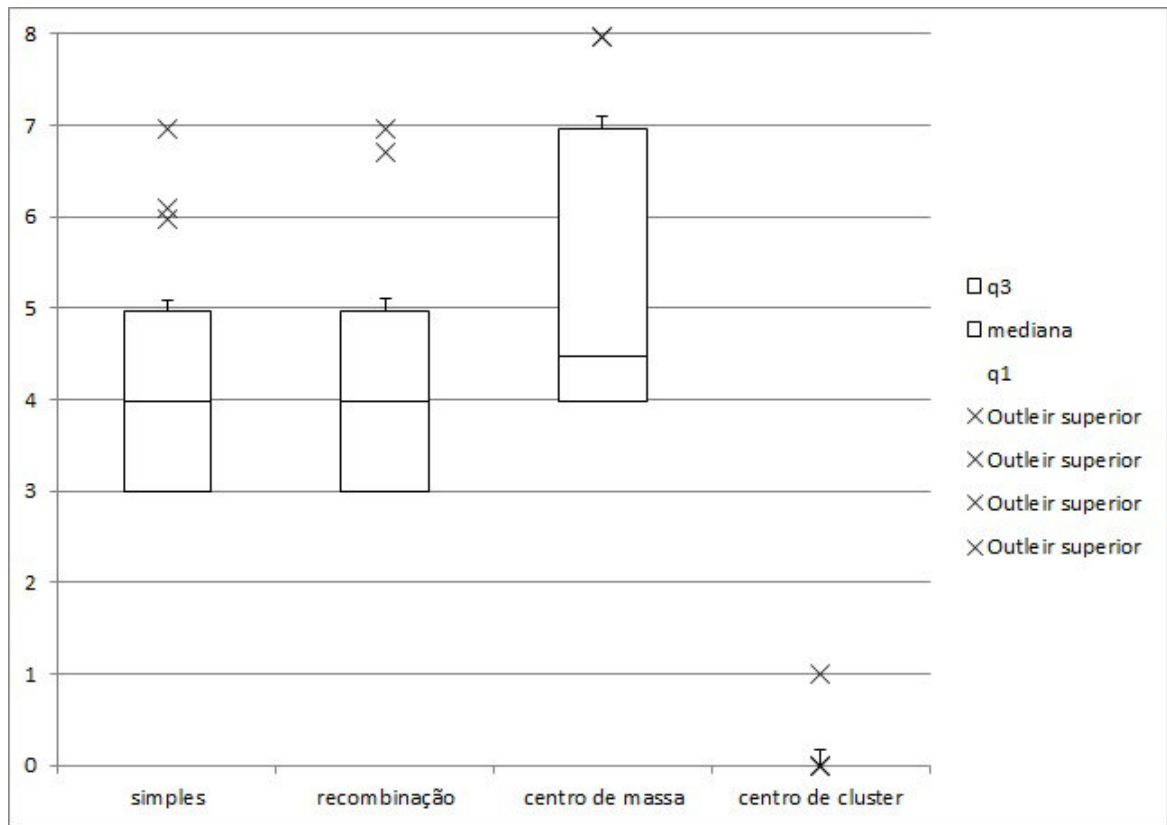


Figura 17: Gráfico Boxplot para Ras com 10 variáveis de decisão

Tabela 5: Rastrigin com 20 variáveis de decisão

Assimilações	Média de melhores resultados	Média de chamadas à FO
<b>Simplex</b>	13,71633082	10019,64286
<b>Recombinação</b>	15,20880654	10017,85714
<b>Centro de massa</b>	19,72672661	10016,60714
<b>Centro de cluster</b>	5,94970375	8642,785714

### 3.3 Resultados obtidos utilizando a função Schwefel

A função *Schwefel* é dada por:

$$f_{Sch}(x) = 418,982887n + \sum_{i=1}^n -x_i \sin\left(\sqrt{|x_i|}\right) \quad (3.3)$$

$$(-500 \leq x_i < 500)$$

$$\min(f_{Sch}(x)) = f(420,9687, \dots, 420,9687) = 0.$$

Os testes realizados foram feitos considerando soluções com 10 e 20 variáveis de decisão.

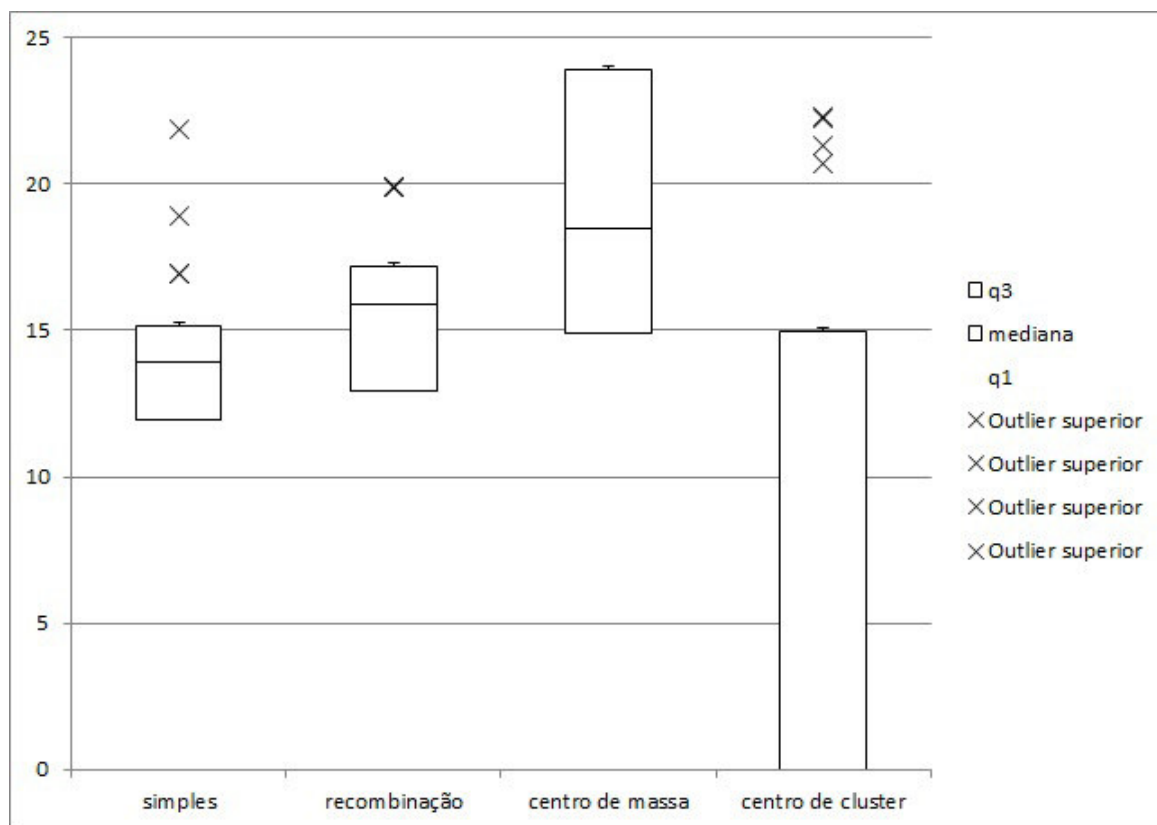


Figura 18: Gráfico Boxplot para Ras com 20 variáveis de decisão

### 3.3.1 Resultados para soluções com 10 variáveis de decisão

Para a função *Sch* com 10 variáveis, a assimilação centro de *cluster* também obteve melhores resultados tanto na comparação das médias (Tabela 6), quanto na comparação no gráfico da Figura 19, sendo visualmente melhor. O centro de massa obteve um desempenho superior aos métodos simples e por recombinação, tanto na comparação das médias, quanto na comparação do gráfico.

Tabela 6: Schwefel com 10 variáveis de decisão

Assimilações	Média de melhores resultados	Média de chamadas à FO
<b>Simple</b>	1043,860761	10009,67857
<b>Recombinação</b>	937,3040665	10009,67857
<b>Centro de massa</b>	867,5688956	10007,67857
<b>Centro de cluster</b>	469,8288041	10000,78571

### 3.3.2 Resultados para soluções com 20 variáveis de decisão

Para a função *Sch* com 20 variáveis, a assimilação centro de *cluster* também obteve melhores resultados tanto na comparação das médias (Tabela 7), quanto comparação do gráfico na Figura 20, sendo visualmente superior. O centro de massa obteve um desempenho

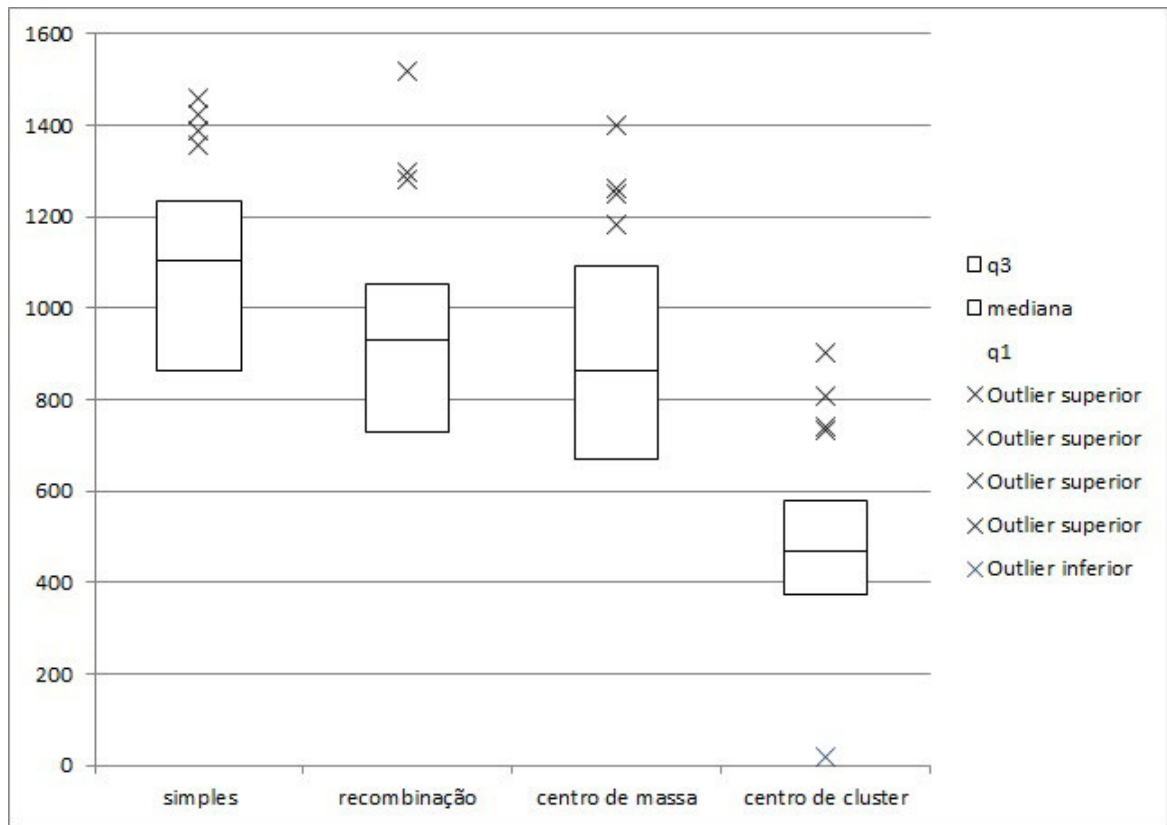


Figura 19: Gráfico Boxplot para Sch com 10 variáveis de decisão

superior ao método simples, porém obteve um desempenho um pouco inferior a assimilação por recombinação, tanto na comparação das médias, quanto na comparação do gráfico.

Tabela 7: Schwefel com 20 variáveis de decisão

Assimilações	Média de melhores resultados	Média de chamadas à FO
<b>Simplex</b>	3357,825964	10019,10714
<b>Recombinação</b>	2993,347708	10019,21429
<b>Centro de massa</b>	3035,996366	10012,25
<b>Centro de cluster</b>	2317,024439	10000,96429

### 3.4 Resultados obtidos utilizando a função Griewank

A função *Griewank* é dada por:

$$f_{Gri}(x) = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \left( \cos\left(\frac{x_i}{\sqrt{i}}\right) \right) \quad (3.4)$$

$(-600 \leq x_i < 600)$   
 $min(f_{Gri}(x)) = f(0, 0, \dots, 0) = 0.$

Os testes realizados foram feitos considerando soluções com 10 e 20 variáveis de decisão.

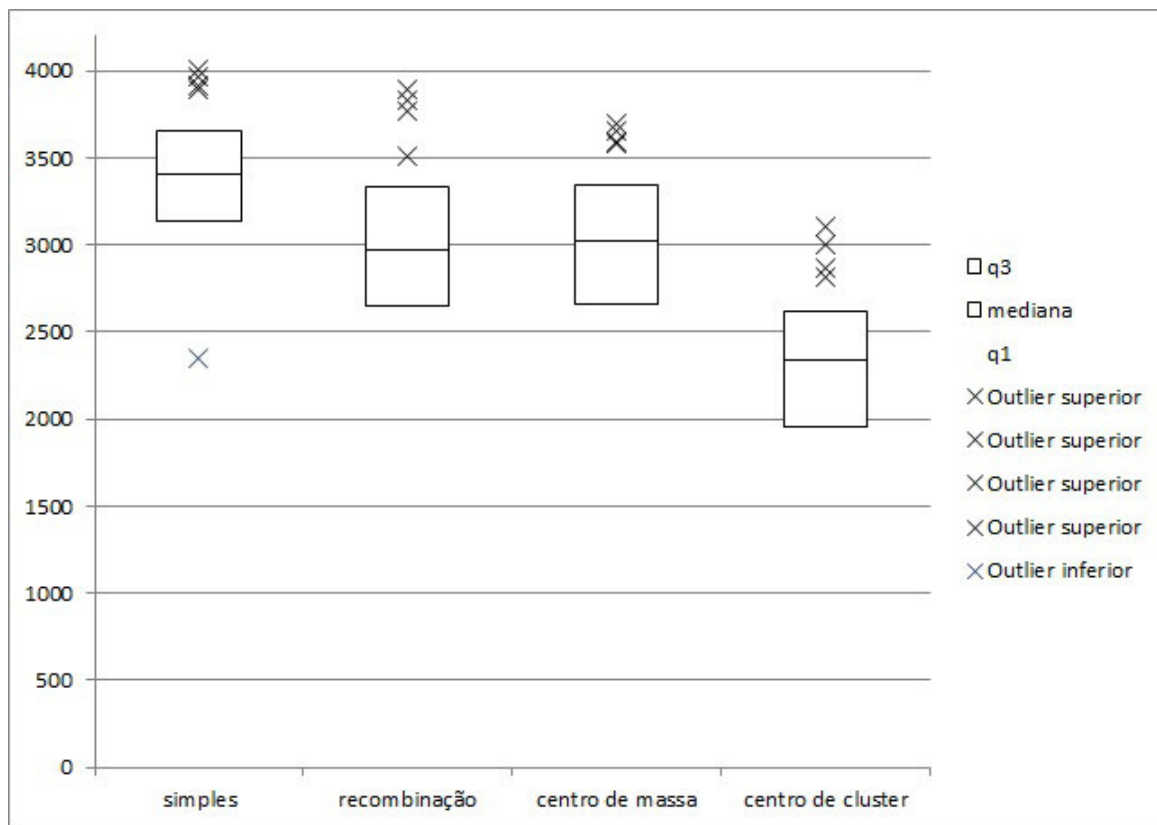


Figura 20: Gráfico Boxplot para Sch com 20 variáveis de decisão

### 3.4.1 Resultados para soluções com 10 variáveis de decisão

Para a função *Gri* com 10 variáveis, a assimilação centro de *cluster* saiu-se relativamente inferior aos outros três métodos, tanto em relação a média de resultados (Tabela 8), quanto na comparação do gráfico da Figura 21, tendo sido visualmente inferior. O centro de massa obteve um desempenho superior apenas ao método centro de *cluster*, porém obteve um desempenho um pouco inferior as assimilações simples e por recombinação, tanto na comparação das médias, quanto na comparação do gráfico.

Tabela 8: Griewank com 10 variáveis de decisão

Assimilações	Média de melhores resultados	Média de chamadas à FO
<b>Simples</b>	0,012936607	9342,75
<b>Recombinação</b>	0,01382	8538,5
<b>Centro de massa</b>	0,017722321	8368,071429
<b>Centro de cluster</b>	0,354938857	10000,85714

### 3.4.2 Resultados para soluções com 20 variáveis de decisão

Para a função *Gri* com 20 variáveis, a assimilação centro de *cluster* saiu relativamente inferior aos outros três métodos, tanto em relação a média de resultados (Tabela 9), quanto na comparação do gráfico da Figura 22, tendo sido visualmente inferior. O centro

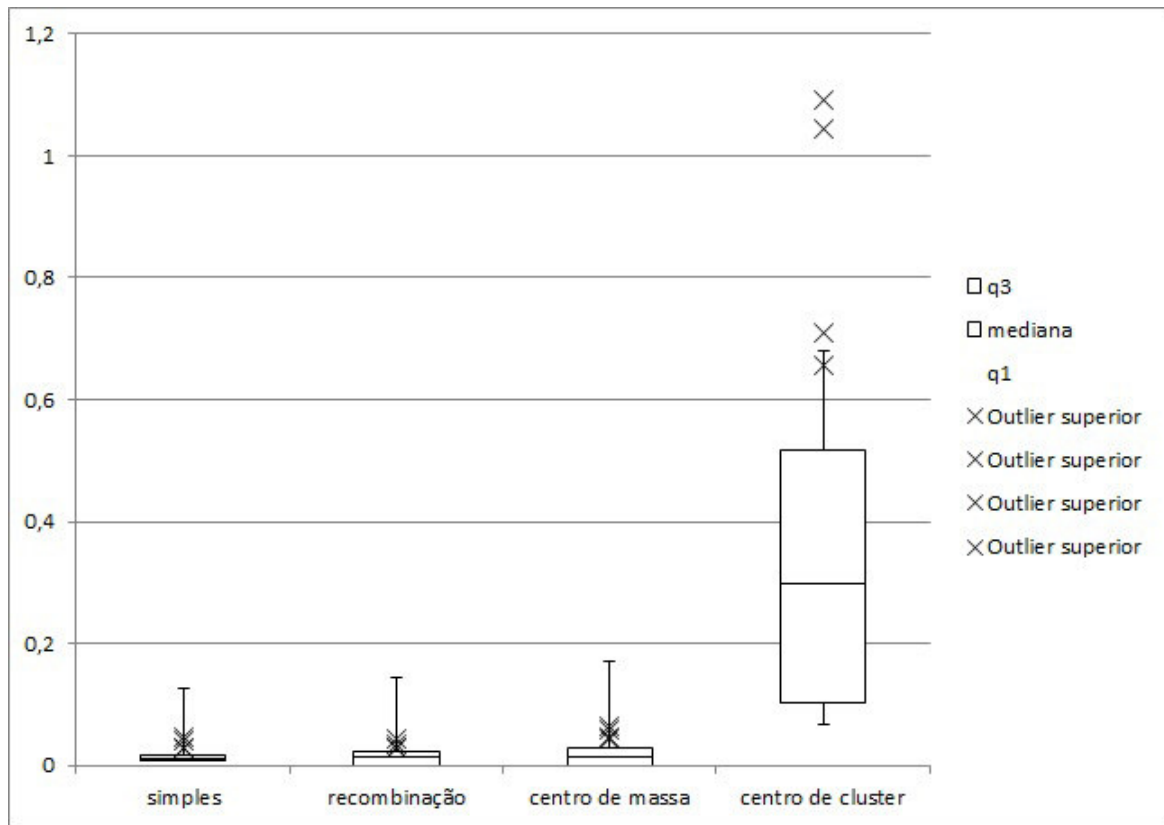


Figura 21: Gráfico Boxplot para Gri com 10 variáveis de decisão

de massa obteve um desempenho superior apenas ao método centro de *cluster*, porém obteve um desempenho bem inferior as assimilações simples e por recombinação, tanto na comparação das médias, quanto na comparação do gráfico.

Tabela 9: Griewank com 20 variáveis de decisão

Assimilações	Média de melhores resultados	Média de chamadas à FO
<b>Simple</b> s	0,00022925	4472,857143
<b>Recombinação</b>	0,000491	3693,964286
<b>Centro de massa</b>	0,000749357	5708,535714
<b>Centro de cluster</b>	1,523352321	10002,53571

### 3.5 Resultados obtidos utilizando a função Langerman

A função *Langerman* é dada por:

$$f_{Lan}(x) = -\sum c_i \left( e^{-\frac{1}{\pi} \|x - A(i)\|^2} \times \cos \left( \pi \times \|x - A(i)\|^2 \right) \right) \quad (3.5)$$

$$(0 \leq x_i < 10)$$

$$\min(f_{Lan}(x)) = -1,4.$$

Os testes realizados foram feitos considerando soluções com 5 e 10 variáveis de decisão.



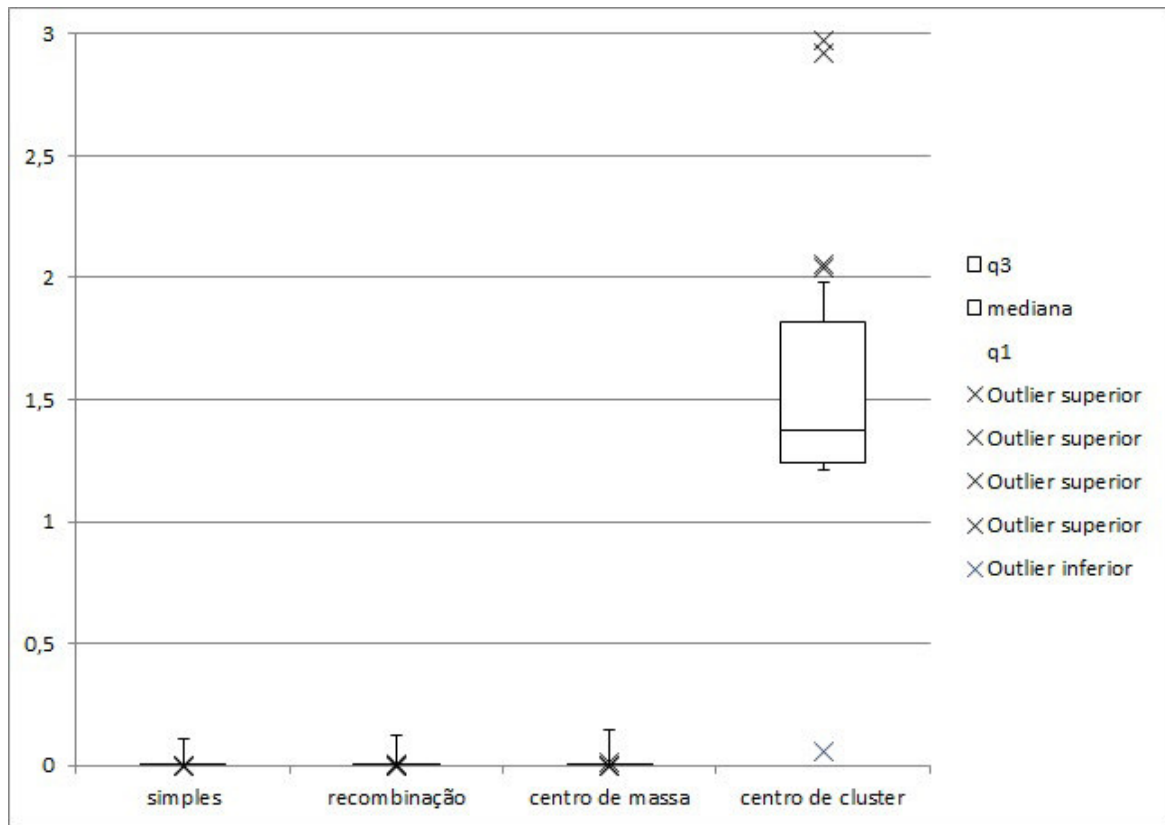


Figura 22: Gráfico Boxplot para Gri com 20 variáveis de decisão

### 3.5.1 Resultados da função Langerman para soluções com 5 variáveis de decisão

Para a função *Lan* com 5 variáveis, a assimilação centro de *cluster* obteve resultados inferior aos outros três métodos apenas em relação a média de resultados (Tabela 10), quanto em relação a comparação no gráfico da Figura 23, o centro de *cluster* se mostrou ter uma variação de resultados inferior aos outros métodos, portanto se mostrando visualmente superior. O centro de massa obteve um desempenho superior aos outros três métodos em se falando da média de valores, porem em relação a comparação no gráfico, ele se mostrou parcialmente semelhante aos métodos simples e por recombinação.

Tabela 10: Langerman com 5 variáveis de decisão

Assimilações	Média de melhores resultados	Média de chamadas à FO
<b>Simple</b> s	-1,360962679	4965,428571
<b>Recombinação</b>	-1,346354714	5623,785714
<b>Centro de massa</b>	-1,391025536	5273
<b>Centro de cluster</b>	-1,337667286	5854

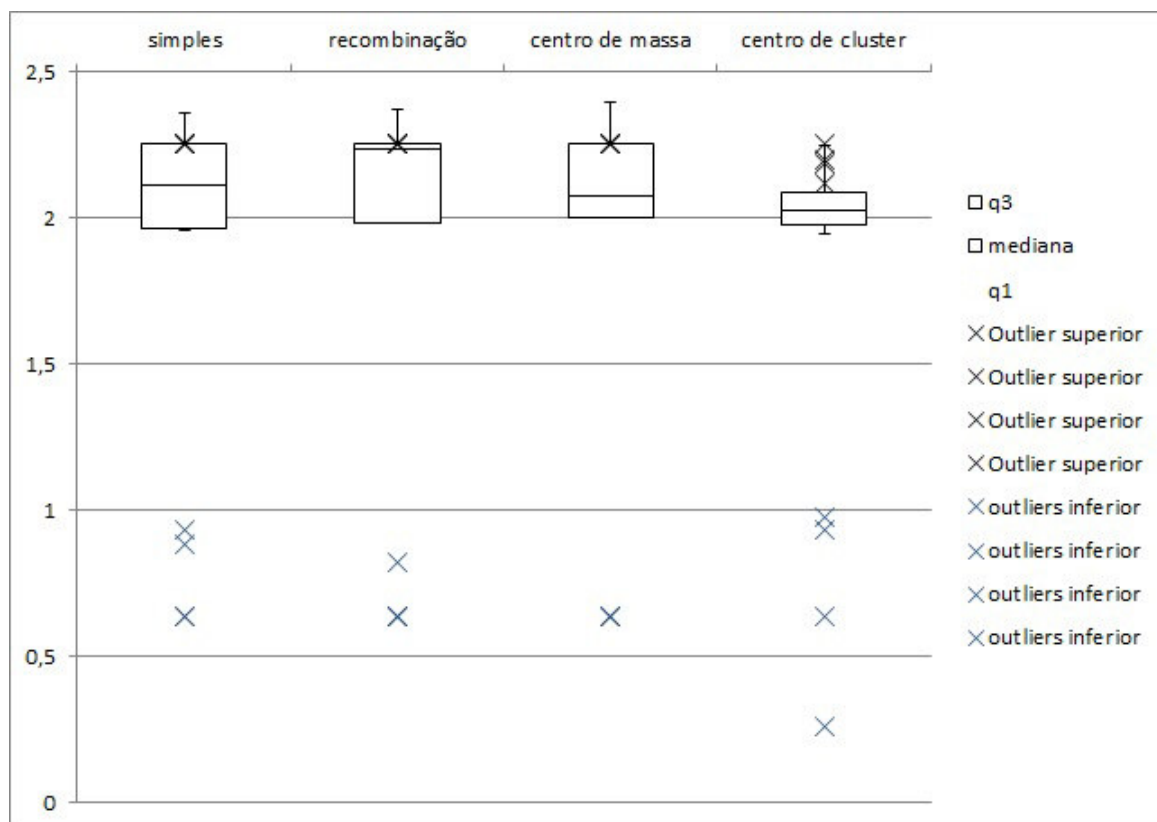


Figura 23: Gráfico Boxplot para Lan com 5 variáveis de decisão

### 3.5.2 Resultados da função Langerman para soluções com 10 variáveis de decisão

Para a função *Lan* com 10 variáveis, a assimilação centro de *cluster* saiu relativamente inferior aos métodos simples, entretanto se saiu um pouco melhor que aos métodos de assimilação por recombinação e centro de massa, isso em relação a média de resultados (Tabela 11), já em relação a comparação no gráfico da Figura 24, o método centro de *cluster* demonstrou ser visualmente melhor que os métodos simples, por recombinação e centro de massa. O centro de massa obteve um desempenho superior apenas ao método recombinação, isso levando em consideração apenas a comparação das médias, considerando a comparação no gráfico, esse método se mostrou semelhante aos métodos simples e por recombinação.

Tabela 11: Langerman com 10 variáveis de decisão

Assimilações	Média de melhores resultados	Média de chamadas à FO
<b>Simple</b> s	-0,809306321	9841
<b>Recombinação</b>	-0,696814071	10006,53571
<b>Centro de massa</b>	-0,701846107	9984,785714
<b>Centro de cluster</b>	-0,706001643	10001,21429

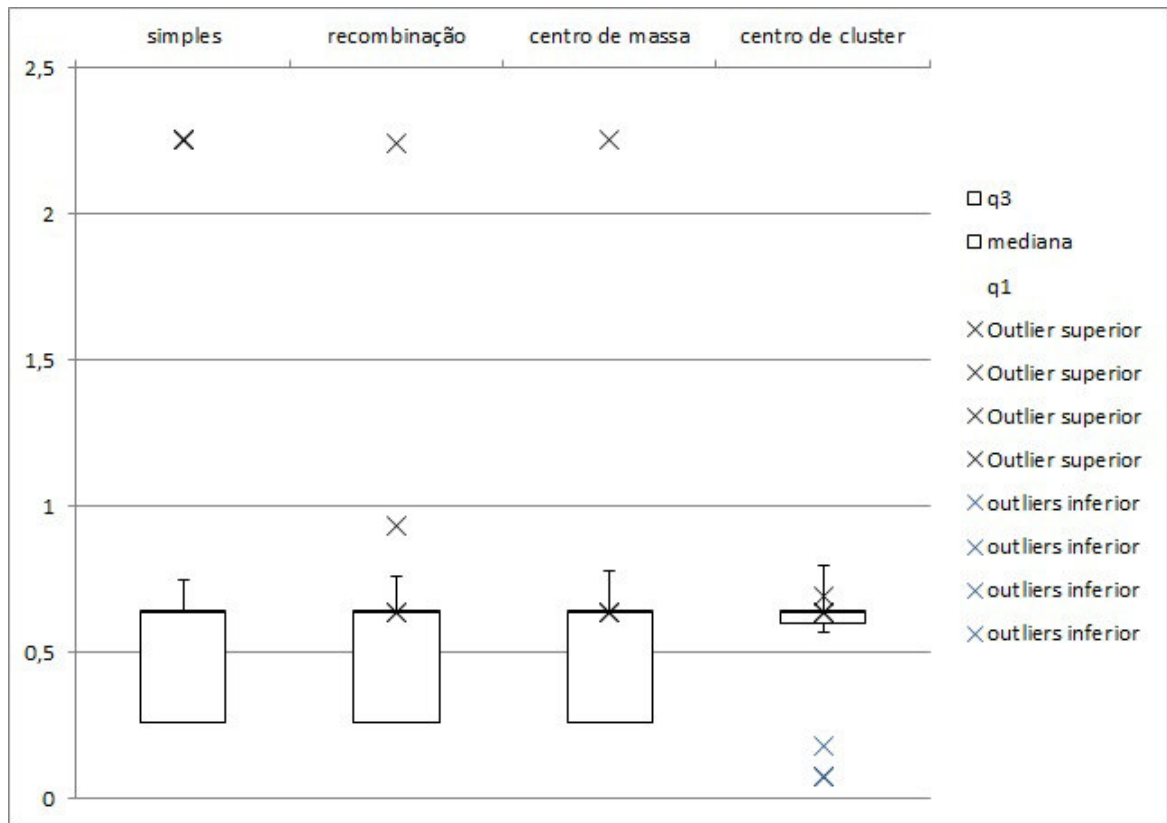


Figura 24: Gráfico Boxplot para Lan com 10 variáveis de decisão

### 3.6 Resultados obtidos utilizando a função Michalewicz

A função *Michalewicz* é dada por:

$$f_{Mic}(x) = - \sum_{i=1}^d \sin(x_i) \times \sin^{2m} \left( \frac{ix_i^2}{\pi} \right) \quad (3.6)$$

$(0 \leq x_i < \pi)$

$min(f_{Mic}(x)) = -4.687 | d = 5, min(f_{Mic}(x)) = -9.66 | d = 10.$

Os testes realizados foram feitos considerando soluções com 5 e 10 variáveis de decisão.

#### 3.6.1 Resultados da função Michalewicz para soluções com 5 variáveis de decisão

Para a função *Mic* com 5 variáveis, a assimilação centro de *cluster* saiu relativamente inferior aos outros três métodos, tanto em relação a média de resultados (Tabela 12), quanto comparação no gráfico da Figura 25, tendo sido visualmente inferior. O centro de massa obteve um desempenho superior apenas ao método centro de *cluster*, porém obteve um desempenho inferior as assimilações simples e por recombinação, tanto na comparação das médias, quanto na comparação do gráfico.

Tabela 12: Michalewicz com 5 variáveis de decisão

Assimilações	Média de melhores resultados	Média de chamadas à FO
<b>Simple</b>	-4,654700893	7119,75
<b>Recombinação</b>	-4,662875607	4749,392857
<b>Centro de massa</b>	-4,632937679	7425,964286
<b>Centro de cluster</b>	-4,6128115	8855,142857

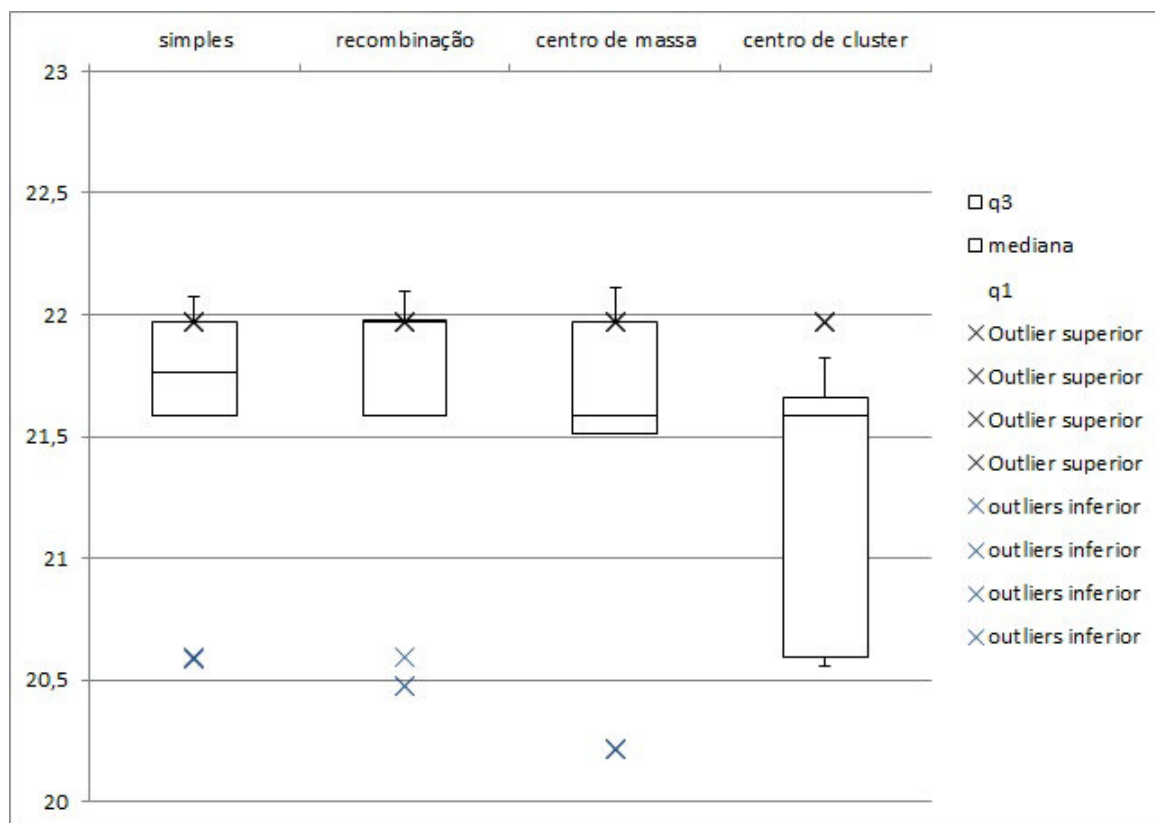


Figura 25: Gráfico Boxplot para Mic com 5 variáveis de decisão

### 3.6.2 Resultados da função Michalewicz para soluções com 10 variáveis de decisão

Para a função *Mic* com 10 variáveis, a assimilação centro de *cluster* saiu relativamente inferior aos outros três métodos, tanto em relação a média de resultados (Tabela 13), quanto na comparação do gráfico da Figura 26, tendo sido visualmente inferior. O centro de massa obteve um desempenho superior aos outros três métodos, em se falando de comparação das médias, porem em relação a comparação no gráfico, este método se saiu inferior ao método simples, sendo semelhante ao método por recombinação e superior ao método centro de *cluster*.

Tabela 13: Michalewicz com 10 variáveis de decisão

Assimilações	Média de melhores resultados	Média de chamadas à FO
<b>Simple</b>	-9,292540857	10007,35714
<b>Recombinação</b>	-9,298895821	10008,32143
<b>Centro de massa</b>	-9,3403175	10004,17857
<b>Centro de cluster</b>	-9,240747429	10000,78571

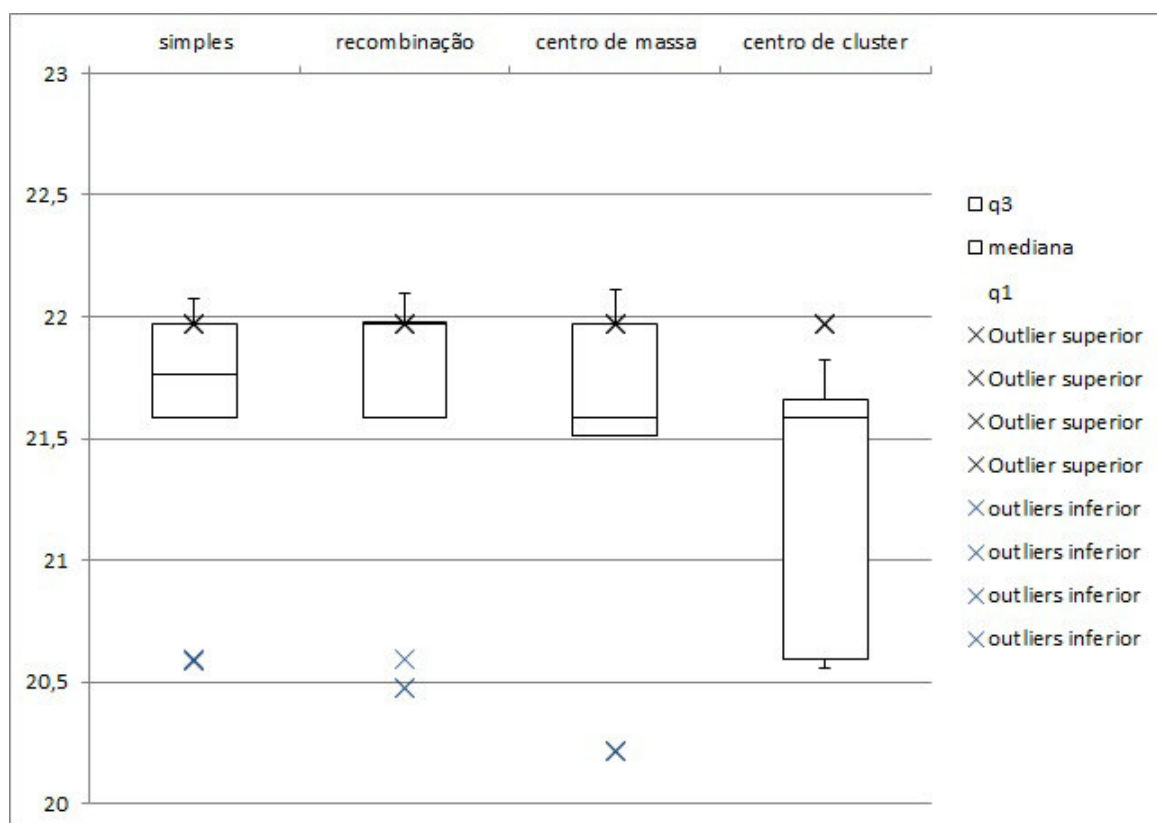


Figura 26: Gráfico Boxplot para Mic com 10 variáveis de decisão

# Conclusão

Este trabalho propõe novos métodos de assimilação para Busca Guiada por Agrupamentos (*Clustering Search*) específico 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 então aplicar operadores de busca local nessas regiões.

Os métodos de assimilação centro de *cluster* e centro de massa mostraram-se eficientes e demonstraram resultados, em vários testes, superiores aos métodos já existentes para otimização contínua, tais como simples e recombinação.

O método centro de *cluster* obteve os melhores resultados nas funções *Ros*, *Ras* e *Sch*, para 10 e 20 variáveis de decisão e levando em consideração tanto na comparação das médias dos resultados, quanto nos gráficos de cada método para com cada função. Para a função *Gri*, o método centro de *cluster* demonstrou-se muito inferior aos método simples, por recombinação e centro de massa, isso considerando 10 e 20 variáveis e também na comparação das médias e gráficos dos métodos para com a função *Gri*. Levando em consideração a função *Lan* e a função *Mic*, não houve uma disparidade tão grande em relação aos resultados, ocorrendo apenas pequenas diferença e distância de valores, tanto na comparação das médias quanto na significância.

O método centro de massa, em comparação com os métodos simples e por recombinação, obteve bons resultados nas funções *Ros* com 10 variáveis, *Sch* com 10 variáveis, para *Sch* com 20 variáveis o método de centro de massa saiu-se melhor apenas que o método simples. Na função *Gri* o método centro de massa saiu-se consideravelmente melhor que o método de centro de *cluster*, tanto para 10, quanto para 20 variáveis e levando em consideração a comparação das médias e quanto no gráfico. Para a função *Lan* e a função *Mic* não houve uma diferença tão significativa de resultados, tanto levando em consideração a comparação de médias quanto na comparação do gráfico.

No geral, esses novos métodos mostraram-se bem competitivos e obtiveram resultados interessantes, demonstrando-se melhores ou equivalentes ao método simples, que é extremamente rápido e de resultados muito bons, em diversas situações. O método centro de *cluster* demonstrou ser melhor que o método centro de massa na maioria das situações, perdendo com uma grande margem apenas na função *Gri*, onde o centro de *cluster* obteve resultados muito inferiores.

Como trabalho futuro pode-se desenvolver novos métodos de assimilação baseados em princípios e técnicas probabilísticas que possam guiar o centro dos agrupamentos para

posições mais promissoras e que gerem melhores resultados no processo de intensificação de busca local. Outro trabalho interessante a ser desenvolvido é gerar um conjunto de técnicas de assimilação que se comporte de maneira diferente e se adaptando a situação de cada agrupamento, tornando o processo de atualização de centro mais eficiente.

# Referências

- BERSINI, H. et al. *Results of the first international contest on evolutionary optimisation (1st ICEO)*. [S.l.]: IEEE Press, 1996. 611–615 p. Citado 2 vezes nas páginas 20 e 31.
- CHAVES, A. A. Uma meta-heurística híbrida com busca por agrupamentos aplicada a problemas de otimização combinatoria. São José dos Campos, 2009. Citado na página 26.
- De Jong, K. A. *An analysis of the behavior of a class of genetic adaptive systems*. Ann Arbor: [s.n.], 1975. 266 p. Citado 2 vezes nas páginas 20 e 31.
- DEB, K. et al. Real-coded evolutionary algorithms with parent-centric recombination. In: IEEE. *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*. [S.l.], 2002. v. 1, p. 61–66. Citado na página 29.
- DIGALAKIS, J.; MARGARITIS, K. *An experimental study of benchmarking functions for genetic algorithms*. [S.l.]: Taylor & Francis, 2002. 403–416 p. Citado 2 vezes nas páginas 20 e 31.
- GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN 0201157675. Citado na página 13.
- HANSEN, N. et al. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, MIT Press, v. 11, n. 1, p. 1–18, 2003. Citado na página 13.
- LINDEN, R. *Algoritmos Genéticos*. 3rd. ed. Rio de Janeiro, RJ, BR: Ciência Moderna Ltda, 2012. ISBN 9788539901951. Citado 2 vezes nas páginas 15 e 16.
- OLIVEIRA, A. C. M. et al. Clustering search. *Pesquisa Operacional*, scielo, v. 33, p. 105 – 121, 04 2013. ISSN 0101-7438. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0101-74382013000100007&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-74382013000100007&nrm=iso)>. Citado 2 vezes nas páginas 13 e 27.
- OLIVEIRA, A. C. M. de. Algoritmos evolutivos para problemas de otimização numérica com variáveis reais. *Monografia apresentada para o exame de qualificação do Curso de Computação Aplicada-Instituto Nacional de Pesquisas Espaciais*, 2001. Citado na página 19.
- OLIVEIRA, A. C. M. de. *Algoritmos Evolutivos Híbridos com Detecção de Regiões Promissoras em Espaços de Busca Contínuos e Discretos*. Tese (Doutorado) — Instituto Nacional de Pesquisas Espaciais, 2004. Citado 5 vezes nas páginas 13, 21, 23, 25 e 31.
- SCHWEFEL, H. P. *Numerical optimization of computer models*. Chichester: Wiley and Sons, 1981. 398 p. Citado 2 vezes nas páginas 20 e 31.